



Universidad
Carlos III de Madrid

Departamento de Ingeniería de Sistemas y Automática

PROYECTO FIN DE CARRERA

DISEÑO Y DESARROLLO DE UN SISTEMA
ANTICOLISION CON SENSORES DE PROXIMIDAD
SIN CONTACTO

Autor: Miguel Ángel Tena García

Tutor: Alberto Jardón Huete

I.T.I.ELECTRÓNICA INDUSTRIAL

LEGANÉS, MADRID

JUNIO DE 2011

TÍTULO: DISEÑO Y DESARROLLO DE UN SISTEMA ANTICOLISION CON SENSORES DE PROXIMIDAD SIN CONTACTO.

AUTOR: MIGUEL ANGEL TENA GARCIA

DIRECTOR: ALBERTO JARDÓN HUETE

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Resumen

En esta memoria se presenta el desarrollo y diseño de un sistema anticolidión mediante sensores sin contacto que comunique con el software de control.

El proyecto es fruto de la necesidad de incrementar la autonomía de los sistemas robotizados. Para ello, los robots, tanto industriales como asistenciales, incorporan cada vez más elementos de sensorización externa, para conseguir una mayor eficacia y flexibilidad a la hora de realizar sus tareas.

Este sistema anticolidión se diseñara para incorporarlo al proyecto ASIBOT. El robot ASIBOT es un “electrodoméstico” móvil, diferente a los robots móviles tradicionales, que puede moverse a través de conectores situados en cualquier punto de la casa, desarrollando diferentes tareas cotidianas de asistencia a personas discapacitadas, como ayuda a comer, beber, aseo personal, manipulación de objetos, etc.

Para incorporarlo al proyecto ASIBOT el sistema anticolidión debe cumplir una serie de requisitos en cuanto a diseño. Se ha ideado que el sistema anticolidión tenga forma circular y que vaya localizado en cada extremo, a modo de pulsera. Debe ser ligera para que no sea un lastre en el movimiento del brazo. Ocupar el menor volumen posible para que no estorbe.

Los sensores que se han utilizado para la implementación son sensores infrarrojos que proporcionan una señal analógica en función de la distancia a la que se encuentra el objeto. Esta señal analógica, enviada por cada sensor que conforma el anillo, es multiplexado y tratado por el microcontrolador y transmitidos a través de Bus CAN.

El diseño software se ha realizado partiendo de la implementación del protocolo CANopen de Microchip® que proporciona un soporte en forma de librerías de programación que facilitan el desarrollo de estos sistemas. El proyecto incluye una guía práctica de estas librerías para el uso y desarrollo de sistemas CANopen.

El sistema anticolidión CANopen realizado se ha puesto en marcha y validado el funcionamiento en laboratorios del departamento de Ingeniería en Sistemas y Automática de la Universidad Carlos III de Madrid.

Agradecimientos

A mi familia y amigos

INDICE GENERAL

Capítulo 1	1
Introducción	
1.1 Motivaciones	8
1.2 Objetivos	9
1.3 Estructura del documento	10
Capítulo 2	11
Estado del arte	
2.1 Asibot	11
2.2 Tecnología anticolisión	15
2.3 Protocolo CAN-Bus	41
2.4 Protocolo CANopen	49
Capítulo 3	68
Diseño	
3.1 Requisitos del sistema	68
3.2 Selección de componentes	69
3.3 Diseño hardware	82
3.4 Diseño software	94
3.5 Configuración software de la comunicación CAN	107
Capítulo 4	117
Ensayos y resultados	
4.1 Pruebas y ensayos sobre el sensor infrarrojo	117
4.2 Pruebas y ensayos sobre el circuito sensor	118
4.3 Ensayos circuito de control	120
4.4 Pruebas en sistemas de control CAN	122
Capítulo 5	125
Conclusiones y futuros desarrollos	
5.1 Conclusiones	125
5.2 Futuras ampliaciones	127
5.3 Análisis crítico	129

Referencias	130
Anexos	131
Anexo 1 Listado de componentes	131
Anexo 2 Presupuesto	132
Anexo 3 Sensor infrarrojo	133
Anexo 4 Acondicionador de señal	135
Anexo 5 Microcontrolador	137
Anexo 6 Multiplexor.	138
Anexo 7 Transceiver	139
Anexo 8 Regulador de tension.	140
Anexo 9 Estructura del mensaje estándar	141
Anexo 10 Esquemáticos	142

INDICE DE FIGURAS

Figura 1	Asibot en anclaje fijo.
Figura 2	Asibot anclado en la silla de ruedas del usuario.
Figura 3	Esquema sensor reed.
Figura 4	Sensor reed.
Figura 5	Sensor magneto inductivo ifm.
Figura 6	Sensor efecto hall y su principio de funcionamiento.
Figura 7	Esquema sensor magnetorresistivo.
Figura 8	Sensor magnetorresistivo.
Figura 9	Sensor Wiegand.
Figura 10	Esquema de sensor inductivo.
Figura 11	Sensores inductivos.
Figura 12	Esquema eléctrico sensor inductivo.
Figura 13	Aplicación del sensor inductivo como final de carrera.
Figura 14	Esquema eléctrico sensor capacitivo.
Figura 15	Partes de un sensor capacitivo.
Figura 16	Tipos de sensores capacitivos.
Figura 17	Aplicación sensores capacitivos para detección de nivel
Figura 18	Aplicación de sensor capacitivo como detector de presencia.
Figura 19	Esquema eléctrico sensor ultrasónico.
Figura 20	Grafica principio de funcionamiento sensor ultrasónico.
Figura 21	Aplicaciones sensor ultrasónico. Detección de bucle, clasificación por alturas y detección de grueso.
Figura 22	Esquema eléctrico sensor de proximidad (emisor y receptor en el mismo cuerpo)
Figura 23	Reserva funcional de un sensor de barrera.
Figura 24	Reserva funcional de un sensor de retrorreflexion.
Figura 25	Reserva funcional de reflexión directa.
Figura 26	Esquema tipos de sensores ópticos.
Figura 27	Funcionamiento sensor de reflexión directa.
Figura 28	Dibujo silla de ruedas con sistema anticolidión
Figura 29	Sistema DAO en uso por un invidente
Figura 30	Tres dedos con sensores ópticos en su cara interna
Figura 31	Sistema CANbus
Figura 32	Niveles de tensión de la señal transmitida
Figura 33	Esquema de sistema conectado a CANbus
Figura 34	CANopen y modelo de referencia OSI
Figura 35	Modelo del nodo.
Figura 36	Modelo de comunicación productor-consumidor en CANopen
Figura 37	Modelos de comunicación punto-a-punto y maestro-esclavo en CANopen
Figura 38	Estructura del identificador de mensajes CAN
Figura 39	Parámetros de un objeto SYNC en CANopen
Figura 40	Máquina de estados de un dispositivo CANopen.
Figura 41	Sensor infrarrojo HDSL-9100
Figura 42	Esquema del sensor y pinout
Figura 43	Grafica Voltaje vs Distancia

Figura 44	Esquema eléctrico acondicionador de señal APDS-9700
Figura 45	Pinout APDS-9700
Figura 46	Pinout del transceiver
Figura 47	Valores de slew-rate en función de Rs
Figura 48	Pinout del multiplexor
Figura 49	Anillo de 16 sensores
Figura 50	Anillo de 24 sensores
Figura 51	Esquema eléctrico del circuito sensor
Figura 52	Versiones del circuito sensor
Figura 53	Esquema circuito de control
Figura 54	Circuito de control
Figura 55	Conexión para la programación del microcontrolador
Figura 56	Directrices para la correcta programación
Figura 57	Esquemático del sistema anticolidión
Figura 58	Sistema anticolidión
Figura 59	Esquema Bloques del circuito sensor
Figura 60	Esquema en Orcad del circuito de programación
Figura 61	Layout del circuito sensor
Figura 62	Layout del circuito de control
Figura 63	Flujograma del programa
Figura 64	Diagrama de actividad
Figura 65	Esquema prueba del sensor infrarrojo
Figura 66	Respuesta del sensor con un objeto a 5mm
Figura 67	Respuesta del sensor con un objeto a 60mm
Figura 68	Respuesta del circuito sensor con un objeto a 5mm
Figura 69	Respuesta del circuito sensor con un objeto a 117mm
Figura 70	Señal pwm que genera el circuito de control
Figura 71	Detección de objetos y visualización led
Figura 72	Sistema anticolidión conectado al sistema de control CAN

INDICE DE TABLAS

Tabla 1	Factores de corrección para sensores ópticos
Tabla 2	Estructura de un diccionario de objetos estándar en CANopen
Tabla 3	Asignación de los identificadores CAN en CANopen
Tabla 4	Códigos de error para SDO Abort Domain Transfer
Tabla 5	Modos de transmisión de PDOs en CANopen
Tabla 6	Estructura de un mensaje de emergencia en CANopen
Tabla 7	Códigos de error para los mensajes de emergencia de CANopen
Tabla 8	Bits del <i>Error Register</i> de los mensajes de emergencia de CANopen
Tabla 9	Trama RTR que el NMT maestro envía a los NMT esclavos
Tabla 10	Mensaje que los NMT esclavos envían al NMT maestro
Tabla 11	Valor del campo <i>state</i> en un NMT Node Guarding Heartbeat
Tabla 12	Estructura de un mensaje de Heartbeat
Tabla 13	Valor del campo <i>state</i> en un mensaje de <i>Heartbeat</i>
Tabla 14	Características eléctricas del HDSL-9100
Tabla 15	Características eléctricas APDS-9700
Tabla 16	Principales características del microcontrolador
Tabla 17	Función de los pines del transceiver
Tabla 18	Gestión de las señales
Tabla 19	Características eléctricas del multiplexor
Tabla 20	Conexiones para comunicación CAN
Tabla 21	Funciones contenidas en la librería <i>pwm.h</i>
Tabla 22	Versión de <i>pwm</i> compatible con el microcontrolador
Tabla 23	Funciones contenidas en la librería <i>adc.h</i>
Tabla 24	Versión de <i>adc</i> compatible con el microcontrolador
Tabla 25	Ficheros de la pila CANopen
Tabla 26	Parámetros del TPDO1
Tabla 27	Parámetros del TPDO2

Capítulo 1

Introducción.

La robótica estudia el diseño y construcción de maquinas capaces de desempeñar tareas repetitivas, tareas en las que se necesita una alta precisión, tareas peligrosas para el ser humano o tareas irrealizables sin la intervención de una maquina [1].

Podemos definir robot como *maquina programable, capaz de manipular objetos y realizar tareas y operaciones antes reservadas solo a las personas.*

La idea de vida artificial viene desde muy atrás y el termino robot se utiliza casi siempre para referirse a humanos mecánicos.

Se ha avanzado mucho en el campo de los robots con inteligencia artificial. Las acciones de este tipo de robots son generalmente llevadas a cabo por motores o actuadores que mueven extremidades o impulsan al robot. El termino robot se ha ido utilizando como un término general que define una maquina o autómatas que imita a un animal, ya sea real o imaginario, pero se ha venido aplicando a muchas maquinas que reemplazan directamente a un ser humano o animal en el trabajo.

Desde la generalización del uso de la tecnología en procesos de producción con la revolución industrial se intento la construcción de dispositivos automáticos que ayudasen o sustituyesen al hombre. Entre ellos destacaron los Jaquemarts, muñecos de dos o más posiciones que golpean campanas accionados por mecanismos de relojería china.

Robots equipados con una sola rueda fueron utilizados para llevar a cabo investigaciones sobre conducta, navegación y planeo de ruta. Cuando intentaron construir robots caminantes, comenzaron con hexápodos y otros tipos de robots de múltiples patas. Estos robots imitaban insectos y artrópodos en funciones y forma. La tendencia se dirige hacia ese tipo de cuerpos que ofrecen gran flexibilidad y han demostrado adaptabilidad a cualquier ambiente. Con más de cuatro piernas, estos robots son estáticamente estables lo que hace que el trabajar con ellos sea más sencillo. Solo recientemente se han hecho progresos hacia los robots con locomoción bípeda.

Recientemente se ha logrado un gran avance en los robots dedicados a la medicina, que han recibido la aprobación regulatoria en América del Norte, Europa y Asia para que este tipo de robots sean utilizados en procedimientos de cirugía invasiva mínima. En la actualidad existen más de 800 robots quirúrgicos en el mundo, con aplicaciones en urología, ginecología, cirugía general, cirugía pediátrica, cirugía torácica, cirugía

cardiaca y ORL. La automatización de laboratorios también es un área en crecimiento. Aquí los robots son utilizados para transportar muestras biológicas o químicas entre instrumentos tales como incubadoras, manejadores de líquidos y lectores.

Otros lugares donde los robots están reemplazando a los humanos son la exploración del fondo oceánico y exploración espacial.

Isaac Asimov introduce por primera vez el término robótica con el sentido de disciplina científica encargada de construir y programar robots. Además plantea que las acciones que desarrolla un robot deben ser dirigidas por una serie de reglas morales, llamadas las tres leyes de la robótica.

1ª Un robot no debe dañar a un ser humano o, por inacción, dejar que un ser humano sufra daño.

2ª Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto si entran en conflicto con la primera ley.

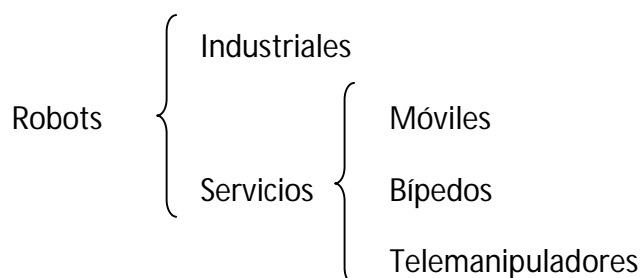
3ª Un robot debe proteger su propia existencia, hasta donde esta protección no entre en conflicto con la primera o la segunda ley.

La función de estas leyes, de la robótica, es una medida de protección para los seres humanos. Las tres leyes de la robótica representan la ética del robot. La concepción de las leyes quería contrarrestar un supuesto "complejo Frankenstein", es decir, un temor que el ser humano desarrollaría unas máquinas que hipotéticamente pudieran rebelarse y alzarse contra sus creadores.

Se formula una ley más, llamada ley cero de la robótica, como corolario filosófico de la primera.

Un robot no puede hacer daño a la humanidad o, por inacción, permitir que la humanidad sufra daño.

No todos los robots son iguales ni se diseñan para desempeñar las mismas funciones, luego se puede realizar diferentes clasificaciones. Nuestra clasificación es la siguiente:



Quizás estos, Robots Industriales y Robots de Servicio, son los dos grandes grupos donde podemos englobar a toda clase de robots.

Los robots son usados hoy en día para llevar a cabo tareas sucias, peligrosas, difíciles, repetitivas o embotadas para los humanos. Esto usualmente toma la forma de un robot industrial usado en las líneas de producción. Otras aplicaciones incluyen la limpieza de residuos tóxicos, exploración espacial, minería, búsqueda y rescate de personas y localización de minas terrestres. La manufactura continua siendo el principal mercado donde los robots son utilizados. En particular, robots similares en movimiento a un brazo humano (robots articulados) son los más usados.

La industria del automóvil ha tomado gran ventaja de esta nueva tecnología donde los robots han sido programados para reemplazar el trabajo de los humanos en muchas tareas repetitivas.

Existen muchas definiciones de robot industrial y todas tienen una característica en común, la aceptación del robot industrial como un brazo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo.

La definición de un robot industrial es un dispositivo de maniobra destinado a ser utilizado en la industria y dotado de uno o vario brazos, fácilmente programable para cumplir operaciones diversas con varios grados de libertad y destinado a sustituir la actividad física del hombre en las tareas repetitivas, monótonas, desagradables o peligrosas.

El trabajo del robot normalmente se limita a pocos movimientos repetitivos de sus ejes. Estos son casi siempre tres para el cuerpo y otros tres para la mano, su radio de acción queda determinado por un sector circular en el espacio donde este alcanza a actuar. Cuando las partes o piezas a manipular son idénticas entre si y se presentan en la misma posición, los movimientos destinados a reubicar o montar partes se efectúan mediante dispositivos articulados que a menudo finalizan con pinzas.

La sucesión de movimientos se ordena en función del fin que se persigue, siendo fundamental la memorización de las secuencias correspondientes a los diversos movimientos. Puede presentarse el caso en el que las piezas o partes a ser manipuladas no se presenten en posiciones prefijadas, en este caso el robot deberá poder reconocer la posición de la pieza y actuar u orientarse para operar sobre ella de forma correcta, es decir se le deberá proveer de un sistema de control adaptativo.

Entre las características que identifican a un robot se encuentran su volumen de trabajo y su configuración

El volumen de trabajo de un robot se refiere únicamente al espacio dentro del cual puede desplazarse el extremo de su muñeca.

Cuando se habla de su configuración, se habla de la forma física que se le ha dado al brazo robot.

Existen cuatro configuraciones posibles:

- Configuración cartesiana, que posee tres movimientos lineales, es decir, tres grados de libertad, los cuales pertenecen a los movimientos sobre los ejes X, Y, Z. El robot cartesiano genera una figura de volumen de trabajo cubica.
- Configuración cilíndrica, que puede realizar dos movimientos lineales y uno rotacional, es decir, presenta tres grados de libertad. El robot de configuración cilíndrica presenta un volumen de trabajo en forma de cilindro.
- Configuración polar, que tiene varias articulaciones y cada una de ellas puede realizar un movimiento distinto: rotacional, angular y lineal.
- Configuración angular. Este tipo de configuración es la que tienen los brazos articulados, que presentan una articulación con movimiento rotacional y dos angulares.

Los robots industriales se pueden clasificar en función del sistema de impulsión, es decir, por el sistema que utiliza para realizar los movimientos.

- Impulsión hidráulica: utiliza algún tipo de fluido, generalmente aceite. Se utiliza para robots grandes, que presentan mayor velocidad u mayor resistencia mecánica
- Impulsión eléctrica: utiliza la energía eléctrica. Se utiliza para conseguir mayor exactitud y repetitividad.
- Impulsión neumática: utiliza aire comprimido. Se utiliza en robots pequeños y limita sus acciones como coger y situar objetos.

Los robots industriales también pueden clasificarse por generaciones.

1ª Generación: repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones de su entorno.

2ª Generación: adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar, detectar y adaptar sus movimientos en consecuencia.

3ª Generación: su programación se realiza mediante el empleo de un lenguaje natural. Posee capacidad para la planificación automática de sus tareas.

En la última década ha aparecido la necesidad de que los robots realicen operaciones distintas de las industriales de producción citadas anteriormente. Para responder a

esta demanda aparece lo que se denomina robótica de servicio. Se define robot de servicio como, un dispositivo electromecánico, móvil o estacionario dotado de uno o varios brazos mecánicos independientes, que opera de manera automática o semiautomática, para realizar servicios útiles al bien estar de los humanos o a su equipamiento, excluyendo las operaciones industriales o de fabricación.

Desde el punto de vista social hay aun muy pocos desarrollos en esta área que hayan impactado en nuestra forma de vivir, pero en los centros de investigación especializados en robótica hay una gran actividad en investigación con este objetivo y, posiblemente en un plazo relativamente inmediato podremos ver robots de servicios operando de forma masiva.

Es evidente, y está en la mente de todos, que los robots son los sistemas que en un futuro van a realizar la mayor parte de las tareas de tipo físico, especialmente las más pesadas.

Como características especiales de los robots de servicio, que los hacen diferentes de los robots industriales, se pueden destacar las siguientes:

- En su aspecto físico los robots dejan de ser un simple brazo, como lo son los robots industriales, para formar una estructura más compleja que puede tener varios brazos y otros dispositivos.
- En general los robots de servicio incluyen dispositivos de locomoción que le dan la capacidad de desplazarse en los diferentes medios en que se vaya a desenvolver. A veces el robot es sencillamente este dispositivo de locomoción.
- La complejidad y falta de estructuración de las operaciones que tienen que realizar, así como la diversidad de objetos que deben manipular, exigen que los robots de servicio estén dotados de un potente sistema sensorial.

En cuanto a la estructura física existen cada vez más exigencias respecto a las características de éstas. Se puede hablar de robots de elevadas dimensiones o de elevada capacidad de carga, lo que requiere estructuras muy robustas, o mini, micro y nano robots, de dimensiones extraordinariamente pequeñas, capaces de introducirse en cavidades pequeñas, para realizar tareas complejas.

Entre los distintos tipos de robots de servicios están los que deben tener la capacidad de desplazarse para situarse en su lugar de trabajo, a veces situado alejado del puesto de control o en un lugar de muy difícil acceso o situado en un entorno incomodo o peligroso para los humanos.

La gran variedad de entornos y situaciones en los que se puede desenvolver un robot exige capacidades de locomoción desarrolladas para ello. Así se puede hablar de

robots terrestres dotados de sistemas de desplazamiento basado en ruedas o, cuando la disposición del medio lo requiere, de patas.

Son robots con grandes capacidades de desplazamiento basadas en carros o plataformas y dotados de un sistema locomotor de tipo rodante. Siguen un camino por telemando o guiándose por la información recibida de su entorno a través de sus sensores.

Estos robots aseguran el transporte de objetos de un punto a otro. Se guían mediante pistas materializadas a través de la radiación fotoeléctricamente. Pueden incluso llegar a sortear obstáculos y están dotados de un nivel relativamente elevado de inteligencia.

Igualmente en sistemas de locomoción hay que citar los robots submarinos con capacidad de desplazarse en el agua. Igualmente los robots con capacidad de volar.

Las necesidades de navegación autónoma de los robots, así como los requerimientos de manipulación de objetos de formas y tamaños diversos exigen también la dotación de sofisticados sistemas sensoriales. Un robot debe ser capaz de reconocer el entorno en el que se va a mover, así como la forma de los objetos que va a manipular. Ello le permitirá además conocer los obstáculos que impiden su movimiento y tomar decisiones para esquivarlos. La visión, igual que en los seres vivos es el sentido más apropiado para este cometido. También es importante destacar las capacidades sensoriales en fuerzas de las que pueden estar dotados los manipuladores de los robots que les permiten realizar tareas con gran destreza y precisión.

Los robots personales son robots capaces de convivir con las personas y de realizar tareas que incidan directamente en su forma de vida. Formalmente formarían parte de los robots de servicios, pero merecen un apartado por su mayor impacto social. Se incluyen en este grupo los robots domésticos, robots de vigilancia, robots de entretenimiento, robots sanitarios, robots en bibliotecas y museos, etc.

La realización de las tareas domésticas, es uno de los problemas importantes con que se enfrentan actualmente las sociedades avanzadas. El desarrollo de equipos capaces de aminorar la carga de trabajo en las casas se ha venido haciendo desde hace más de cincuenta años con la invasión de electrodomésticos, bastante de ellos imprescindibles en toda vivienda. Pero, aunque a algunos se les haya denominado robots, todos tenemos la conciencia de que esa denominación no corresponde a sus prestaciones. Un robot domestico debe realizar las tareas de la casa con total autonomía sin más intervención de las personas que, como mucho, para realizar una programación previa de sus actividades.

El primer sistema que responde a las características de un robot, que ha penetrado en el mercado de los equipos domésticos, ha sido el robot aspiradora. Estos son programables, están sensorizados, utilizan algoritmos que les permiten cubrir todo el

suelo del recinto a aspirar, se desplazan siguiendo el límite de las paredes y alrededor de los muebles y son capaces de dirigirse al punto de alimentación eléctrica cuando sienten que sus baterías están bajas. Además sus costes están en línea con cualquiera de los electrodomésticos de funcionamiento manual.

Otras funciones susceptibles de ser robotizadas en un futuro son las de manipulación inteligente de objetos domésticos, como vajillas u otros utensilios, ordenación de armarios o estanterías, etc. Estos robots todavía no existen, aunque en algunos centros de investigación se está trabajando en su desarrollo. Deben contar, además de con brazos manipuladores con sistemas de locomoción, así como con métodos sencillos de programación de sus actividades. En este sentido, los métodos de aprendizaje automático e imitación son los que quizás tengan mayor futuro.

Una extensión de este tipo de robots, aún en desarrollo en ciertos centros de investigación, es el robot de asistencia a ancianos o discapacitados. Se estima que en el año 2025 el 20% de la población europea será mayor de 65 años, muchos de ellos con necesidad de ayuda para desarrollar una vida normal.

Es claro que la solución definitiva a este problema se tendrá que realizar con la introducción masiva de robots que ayuden a personas dependientes en la realización de sus tareas cotidianas como vestirse, bañarse, realizar su aseo personal, comer, etc. Lógicamente, el nivel de interacción de estos robots con los humanos tiene que ser muy fuerte y ello exige un levado sistema cognitivo cuyo desarrollo se encuentra en la actualidad lejos de poderse considerar acabado.

Es importante, para completar esta visión de los robots personales, hablar de la asistencia y cuidado a enfermos y, en general, de las oportunidades que los robots ofrecen en el campo de la sanidad.

En asistencia a enfermos existen ya prototipos de robots que realizan rondas de vigilancia en hospitales llevando los medicamentos que los enfermos deben ingerir en cada momento. Igualmente pueden monitorizar y transmitir a un puesto remoto de control las variables sobre el estado del paciente.

Pero en lo que vislumbra un futuro verdaderamente prometedor es en el campo de la cirugía. Esta ha prosperado en los últimos años gracias a los medios tecnológicos de la que se ha dotado. Entre los principales figuran los que introducen elementos robóticos como los de laparoscopia y, en general, la denominado cirugía mínimamente invasiva.

Ya en la década de los 70 se decía que en el desarrollo de la robótica se iban a suceder tres fases: el nacimiento e infancia, que reflejaba la situación de la misma en aquellos tiempos, un crecimiento progresivo hasta alcanzar una madurez, y la explosión. En esta última fase, se decía, se iba a producir un uso masivo de los robots en todas las

actividades de la vida con la pretensión de que ellos hicieran todas las labores productivas y de servicio.

Hay temas para hacer esto realidad que aun requieren un desarrollo tecnológico importante, aunque ya se han dado bastantes pasos en esa dirección. Se trata del desarrollo de lo que podríamos denominar los cerebros de los robots. Estos deben ser procesadores con capacidad de realizar las mismas funciones que los cerebros humanos. Pero aun hace falta más, por ejemplo, el sistema emocional. La interacción con humanos exige dotar a los robots de un sistema de estas características.

La línea de investigación, quizás, más prometedora, es aquella que configura estos cerebros como redes de neuronas artificiales que interactúan entre ellas y que pueden aprender conductas de comportamiento en función de estímulos.

También es importante, en un contexto de vaticinio del futuro de los robots, citar alguna de las aportaciones que otras disciplinas, como la nanotecnología o la biotecnología, pueden realizar a la robótica en las próximas décadas. La nanotecnología permitirá la construcción de sistemas del tamaño de una molécula dotados de movimientos y que, unidos entre sí formaran colonias que operan conjuntamente. Estos robots trabajando conjuntamente podrán ser usados como infatigables trabajadores construyendo o montando todos los objetos que necesitemos. De manera similar a las hormigas, se podrá, realizar exploraciones de zonas desconocidas.

Nano robots de características similares podrán ser utilizados en biotecnología, se podrán construir órganos artificiales para implantes o vehículos diminutos que se introduzcan en los cuerpos para realizar cirugías o reconocimientos de determinados órganos, o para depositar, o incluso ser, la medicación microscópica para curar determinadas células enfermas.

1.1 Motivación

Basándose en las actuales líneas de investigación de la robótica, que están orientadas a la robótica de servicio, Roboticslab del Departamento de Ingeniería de sistemas y Automática de la Universidad Carlos III de Madrid han creado ASIBOT, que es un robot manipulador escalador portátil.

La motivación final es la realización un proyecto que permita al ASIBOT desarrollar de manera más segura sus tareas, evitando colisiones indeseadas. Se trata de un sistema multisensorial capaz de detectar objetos cercanos a la zona de actuación y capaz de actuar ante ellos.

ASIBOT es un prototipo de investigación y que va incorporando distintos sistemas y componentes, cuyo bus interno de control es CANopen y está preparado para futuras ampliaciones.

CANbus es un sistema estandarizado de comunicación serie muy potente y que permite una gran flexibilidad a la hora de sustituir e introducir nuevos componentes a la red. Este sistema sólo cubre las necesidades hardware de sistema, por lo que es necesario introducir un protocolo de comunicaciones a nivel software. Existen varios protocolos estandarizados de nivel software como son el CANopen, DeviceNet, SAE J1939 o CANKingdom.

Además del bus, ASIBOT dispone de un bus de potencia de 24V y 5V que a lo largo de su estructura, siendo accesible en los extremos y en los eslabones centrales. Esto permite conectar dispositivos adicionales.

1.2 Objetivos.

ASIBOT es un robot de asistencia personal a ancianos y discapacitados. Este robot lo dirige el usuario y por tanto es necesario un sistema que permita al usuario controlar el movimiento del extremo en un espacio reducido y rodeado de objetos.

Uno de los objetivos de este proyecto es realizar un estudio exhaustivo de los diferentes tipos de sensores de proximidad sin contacto que se encuentran en el mercado y que reúnen las características adecuadas para la aplicación requerida.

Otro objetivo es el estudio de diferentes sistemas anticollisión ya existentes para barajar diferentes alternativas de diseño para un sistema anticollisión que permita al usuario manipular los objetos deseados sin dañar otros que estén cerca o colisionar con el entorno.

El sistema anticollisión debe detectar los objetos que se encuentra alrededor de la pinza, con la que manipulará los objetos, en un radio ajustable de 20cm y en caso de peligro de colisión debe enviar a tiempo la información para poder detener el movimiento del ASIBOT para evitarla. Se debe fijar una distancia de seguridad, dependiendo de la velocidad de movimiento y cuando se traspase esta distancia de seguridad tomar la decisión de STOP.

Para la implementación del protocolo CANopen se usarán componentes de Microchip por su gran variedad y disponibilidad en el mercado, además de por la existencia de herramientas de diseño muy potentes y entornos de desarrollo adecuados para diseñar sistemas CAN.

1.3 Estructura del documento

Un proyecto de ingeniería, en este caso el diseño e implementación de un sistema anticolidión para ASIBOT, se divide en varias fases:

- Estudio previos. En esta fase se plantean los objetivos, motivaciones y necesidades que el diseño debe cumplir. Se define un modelo conceptual y se analiza su viabilidad. Se estudian artículos, proyectos existentes y relacionados. Se estudia la tecnología y componentes, necesarios para su construcción, que existen en el mercado, con el fin de encontrar el que más se adapte a las necesidades del sistema.
- Diseño y simulación. En esta fase se realiza un modelado del sistema que luego irá insertado en el ASIBOT, siguiendo el modelo conceptual definido en la fase anterior. Una vez obtenidos los resultados esperados se pasa a la última fase.
- Implementación y ensayos. Se trata de construir el sistema real que irá montado en el ASIBOT y realizar las pruebas pertinentes para comprobar su correcto funcionamiento y que por tanto se cumplen las expectativas de diseño y simulación.

La memoria del proyecto de fin de carrera también forma parte del proyecto de ingeniería y debe estar estructurada.

La memoria de este proyecto de fin de carrera está estructurada en cinco capítulos, la bibliografía y anexos.

- Capítulo 1. En el primer capítulo se realiza una introducción a la robótica y tipos de robots, así como su evolución. Además se describen los objetivos y motivaciones del proyecto, cumpliendo con la primera fase de un proyecto de ingeniería.
- Capítulo 2. En este capítulo se explica con detalle la función y funcionamiento del ASIBOT, las características del protocolo de CANopen y CANbus. Se realiza un comparativo de los diferentes tipos de sensores de proximidad y un estudio de diferentes sistemas anticolidión ya implementados.
- Capítulo 3. Aquí se explican detalladamente los requisitos que debe cumplir el sistema anticolidión. Se explica el desarrollo completo del proyecto, tanto hardware como software. Sin duda es el capítulo más importante del proyecto ya que explica con detalle el diseño e implementación del sistema anticolidión. Se realiza la selección de los componentes más adecuados y que cumplen con los requisitos y necesidades del sistema anticolidión.
- Capítulo 4. Aquí se describirán los resultados y conclusiones de prácticas y experimentos del sistema anticolidión diseñado.
- Capítulo 5. Se propondrán futuras mejoras para el sistema anticolidión.

Capítulo 2

2.1 Asibot

Actualmente la robótica de servicios es la línea de investigación prioritaria en el campo de la robótica y automatización. Los robots móviles actuales no están adaptados para ser utilizados en entornos domésticos, debido a su volumen y falta de maniobrabilidad en un entorno tan complejo como son estos escenarios por lo que se está trabajando en nuevas línea de diseño. La aplicación de los robots a tareas de servicios (asistencia personal, educación, tareas sociales) hace que su diseño sea de suma importancia.

Existen muchos proyectos de desarrollo de robots humanoides adaptados a entornos domésticos o de oficinas que trabajen como robots de servicio, puesto que pueden llegar a tener mayor maniobrabilidad.

Los robots humanoides tiene el problema de que debido al alto número de grados de libertad necesitan un actuador, motor o reductor para cada grado de libertad, lo que los convierte en maquinas pesadas. Existen líneas de investigación basadas en técnicas de control dinámica pasiva para reducir el número de actuadores y con ello el peso.

Una alternativa a los robots humanoides es el ASIBOT [2]. Es un robot diferente a los robots móviles tradicionales ya que puede moverse a través de conectores situados en cualquier punto de la casa, desarrollando diferentes tareas cotidianas de asistencia a personas discapacitadas, como ayuda a comer, beber, aseo personal, manipulación de objetos, etc.

Hasta la fecha la tecnología utilizada en el hogar ha venido en los electrodomésticos como lavavajillas, hornos, etc. Más tarde se introdujeron sensores y actuadores, permitiendo cierta automatización, el control de la temperatura, detección de intrusos e incendios. Pero todos estos elementos son estáticos, están instalados en la casa y permanecerán en ella durante toda su vida útil realizando siempre la misma tarea.

El siguiente paso en la introducción de nuevas tecnologías en el entorno domestico es introducir elementos inteligentes y móviles en la casa, es decir, robots. Así mismo, otro paso adelante consistiría en extrapolarlo fuera del hogar, la oficina, el supermercado, etc., de forma que el usuario disponga de las ayudas modulares también fuera de su hogar adaptado.

Existen diversas propuestas para el desarrollo de sistemas de asistencia modulares, a personas mayores y discapacitadas, para el trabajo y el hogar. Consisten fundamentalmente en integrar bajo una red de comunicaciones demótica, tres tipos de módulos, compartiendo soporte físico y protocolo:

- Dispositivos domóticos estándar, que implementan control de iluminación, ventanas, persianas, puertas, etc.
- Electrodomésticos de gama blanca y marrón, dotados de conectividad. Se incluyen tecnologías de la información y comunicación en el hogar ADSL, sistemas de tele-asistencia médica, tele-conferencia, etc.
- Ayudas técnicas basadas en Robots Asistenciales (RA), como sillas de ruedas autónomas y sistemas de asistencia personal, RA sobre plataforma móvil, RA portátil ASIBOT, etc.

La utilidad de cualquier ayuda, sea robótica o no, está estrechamente relacionada con su facilidad de uso, que vendrá determinada por la complejidad del sistema y la interface de usuario.

La característica fundamental que debe asegurar un robot asistencial, es la de posibilidad de aumentar las capacidades del usuario, gracias a la utilización del sistema sensorial y al sistema locomotor, que ejecutará las acciones deseada sobre el entorno, cubriendo así tanto las deficiencias perceptivas, como las deficiencias motoras del usuario. Si todo esto se puede desplazar a donde vaya el usuario, se puede considerar que el sistema es portátil. Para que se pueda desplazar el robo tendrá su propio sistema locomotor o usará el medio de locomoción que el usuario, generalmente la silla de ruedas.

ASIBOT presenta cinco grados de libertad y una configuración cinemática simétrica, lo que le permite anclarse por ambos extremos. Tiene un mecanismo para conectarse a la pared o a la silla de ruedas y una pinza. El robot tiene una estructura de aluminio en sus diferentes articulaciones y los eslabones son de fibra de carbono. Estos dos eslabones centrales contienen los equipos electrónicos y la unidad de control del brazo. De esta forma, el robot con 12 Kg, es completamente autónomo salvo que precisa de conexión exterior para su alimentación. Los actuadores se han diseñado de forma que servomotor, freno y reductor quedan integrados formando cada eje motriz extremadamente compacto y ligero. La energía se recoge del conector eléctrico ubicado en el centro del anclaje mecánico.

ASIBOT puede moverse con gran precisión y fiabilidad entre los conectores de anclaje. Los conectores pueden ser alojados en la pared o bien se instalan sobre un raíl en determinadas localizaciones con el objetivo prolongar el rango de acción. Para que el sistema ASIBOT sea totalmente operativo, se precisa cierta adaptación en la instalación de conectores de anclaje en determinados lugares de paso y trabajo haciendo posible la transición de un lugar a otro de la casa o desde una pared a una silla de ruedas o viceversa.

Este alto grado de flexibilidad tiene una gran implicación en el cuidado de personas discapacitadas o ancianos con problemas de movilidad en las extremidades superiores.

La modularidad del sistema robótico hace posible que este crezca tanto como sea el nivel de discapacidad de la persona que lo vaya a utilizar. Los conectores permitirán al robot desplazarse de un lugar de trabajo a otro de forma autónoma siguiendo directivas de movimientos pre-programados según las necesidades del usuario. Se concibieron tres clases distintas de anclajes o DS:

- Fijas: a la pared, techos, suelo, muebles u otros lugares de la casa donde sean necesarios para la tarea que se vaya a realizar, tales como colocar platos en una mesa o dar de comer a una persona. En la figura1 se muestra ASIBOT en un anclaje en la mesa.

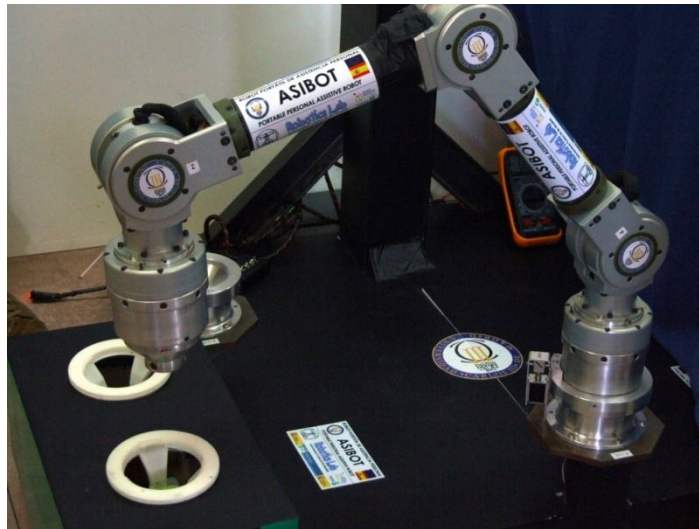


Figura1. Asibot en anclaje fijo

- Móviles. Sobre un rail permiten al robot desplazarse distancias largas a mayor velocidad
- Si se coloca un DS en un rail montado en la silla de ruedas, el usuario podrá usar el robot como los clásicos RA MANUS y RAPTOR. En la figura 2 podemos observar a ASIBOT en la DS de la silla de ruedas de un usuario.

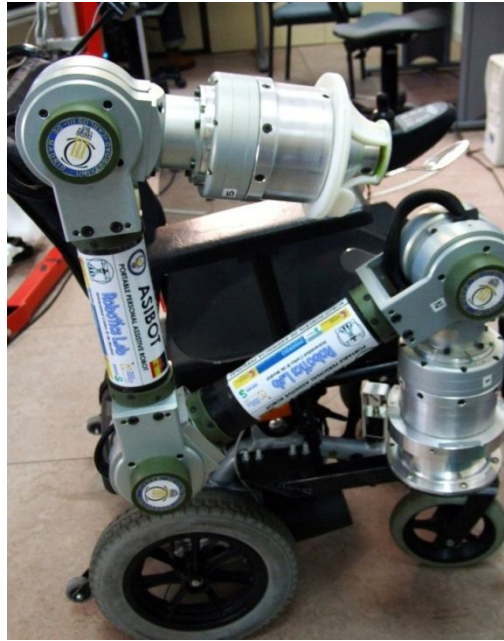


Figura2. Asibot anclado en una DS

Habr  una DS especial en un lugar de la casa para permitir la transici n del robot desde la silla a la pared de DS fijas y viceversa.

ASIBOT podr a clasificarse en distintos tipos de robots. Cada extremo aloja un sistema de anclaje los que le permite trabajar en diferentes posiciones, all  donde tenga una DS y poder moverse de una a otra, como los robots escaladores. Dispone de una garra formada por tres dedos retractiles, que se esconden para realizar el anclaje, y que le permite, desde su posici n de trabajo, realizar sus operaciones principales que son coger y mover cosas o posicionar herramientas, igual que los robots manipuladores. Sin embargo debido a su car cter auto-contenido, su bajo peso y su capacidad de anclarse al DS de la silla del usuario, lo denominaremos RA (Robot Asistencial) port til.

Las principales tareas de AISBOT est n relacionadas con las tarea domestica. No es necesario un alto grado de precisi n ya que durante el proceso de dise o se decidi  que mientras se realiza la tarea de dar de comer o las tareas de afeitado, el robot lo  nico que tiene que hacer es presentar la cuchara o la maquinilla de afeit r, o el cepillo de dientes al usuario. Durante estas tareas es muy importante el control de las distintas trayectorias del brazo y sus perfiles de velocidad y aceleraci n, ya que el robot se mover  muy cerca del usuario. Si el robot se mueve con la cuchara llena de comida, ser  crucial el control de la orientaci n del extremo para evitar que esta se caiga de la cuchara.

As  el robot permite acercar objetos, coger y mover utensilios usados para las actividades de la vida diaria hasta un plano de trabajo pr ximo al usuario, y este con la actividad residual del cuello puede completar el uso.

En definitiva se ha logrado integrar en un diseño innovador y portátil las funcionalidades de robot escaladores y brazo manipuladores con un propósito asistencial para personas discapacitadas, y anclando el robot en la silla de ruedas del usuario el ámbito de operación no se limita al entorno.

En la realización de las distintas tareas de ASIBOT es necesario un control de las distancias y para ello se quiere diseñar un sistema anticolidión que permita al robot evitar colisiones indeseadas contra otros objetos cercanos al que se dispone a manipular o que se encuentre en las trayectorias seguidas por el brazo hasta su objetivo.

Para la implementación de este sistema anticolidión se instalaran sensores gobernados por un microprocesador o PIC. Los sensores en presencia de un objeto proporcionaran una señal de salida en función de la distancia al objeto. Esta señal será interpretada por el microprocesador. Una vez rebasada la distancia de seguridad que se haya fijado, el microprocesador considerara peligro de colisión y enviara por CANbus la variable correspondiente al sensor que ha detectado el objeto dentro de la distancia de peligro, para que ASIBOT la interprete y tome la decisión de para su movimiento.

Un factor importante a la hora de la implementación del sistema anticolidión es el tipo de sensores. Para seleccionar el más adecuado es necesario realizar un estudio de la tecnología anticolidión que podemos encontrar en el mercado y realizar la comparación de diferentes tipos de sensores de proximidad.

Otro factor importante es la ubicación de los sensores. Puesto que se trata de un brazo robot se pensó que el sistema anticolidión debería tener forma de pulsera o anillo. ASIBOT posee una pinza con tres dedos que le permite manipular objetos y herramientas y puesto que es esta parte del robot la que entrara en contacto con los objetos se pensó que la ubicación más adecuada para nuestro anillo anticolidión fuera el extremo de ASIBOT, ya que al estar tan cerca de la pinza le permitiría localizar objetos en la trayectoria que realiza el brazo o que estén cerca del objeto que la pinza quiere manipular.

2.2 Tecnología anticolidión

Los sensores de proximidad conforman un amplio grupo de dispositivos utilizados para detectar la presencia de objetos o personas en torno a una zona delimitada, pero también para determinar la presencia de elementos de un determinado material en la zona de trabajo.

Los sensores aportan la información básica de campo, son los que incorporan las variables de control o las magnitudes de referencia a un sistema del que se desea

obtener controlabilidad o información de estado en su funcionamiento. Junto a los actuadores, son los dispositivos que permiten la interactividad necesaria entre la planta y el sistema de control para que el conjunto funcione adecuadamente. Los sensores indican el estado de ciertas variables medibles y los actuadores actúan sobre el proceso supervisado en función de la gestión que se establezcan para cada situación.

De entre los distintos tipos de sensores destacan los sensores de proximidad por varias razones. La primera de ellas es que se trata de un amplio grupo de dispositivos, heterogéneo, compuesto por sensores de diferentes tipos, con principios de funcionamiento diferentes y aplicaciones distintas. Así, pertenecen a este grupo tipos tan distintos como pueden ser los finales de carrera (interruptores mecánicos, con detección por contacto físico) o los sensores optoelectrónicos láser (sensores de detección sin contacto). El nexo que los mantiene unidos como grupo es que se emplean para detección de proximidad de objetos o personas. Otra de las razones que explican la importancia de este tipo de sensores es que la detección de proximidad es una de las aplicaciones más extendidas y más versátiles. Con la información de detección de proximidad de objetos se puede precisar el estado predefinido de un sistema (máquina, cadena de producción, sistema de transporte, o cualquier dispositivo que incorpore detección de proximidad) y permite que el sistema actúe según ha sido establecido.

La meta de este proyecto de fin de carrera es la realización de un sistema anticolidión, para el brazo ASIBOT. Con este sistema anticolidión se pretende evitar colisiones indeseables. Debemos tener un sistema anticolidión con un rango de detección variable que nos permita acercarnos a objetos todo lo que deseemos. Un sensor de proximidad es capaz de detectar objetos en un rango muy amplio de distancias y puesto que nuestro sistema se utilizara en espacios cerrados, es necesario que aunque nuestro sensor detecte distintos objetos no los considere una amenaza, por lo que deberemos establecer un umbral a partir del cual se considerara que la colisión es inminente y el sistema deberá detenerse para evitarla.

A la hora de elegir el sensor debemos tener en cuenta una serie de restricciones y las características de los distintos tipos de sensores que podemos encontrar en el mercado.

Para nuestro sistema anticolidión utilizaremos un sistema multisensor, que se trata de un sistema compuesto de varios sensores, en este caso, de idénticas características.

Podemos distinguir dos tipos de sensores [3] según l tipo de salida:

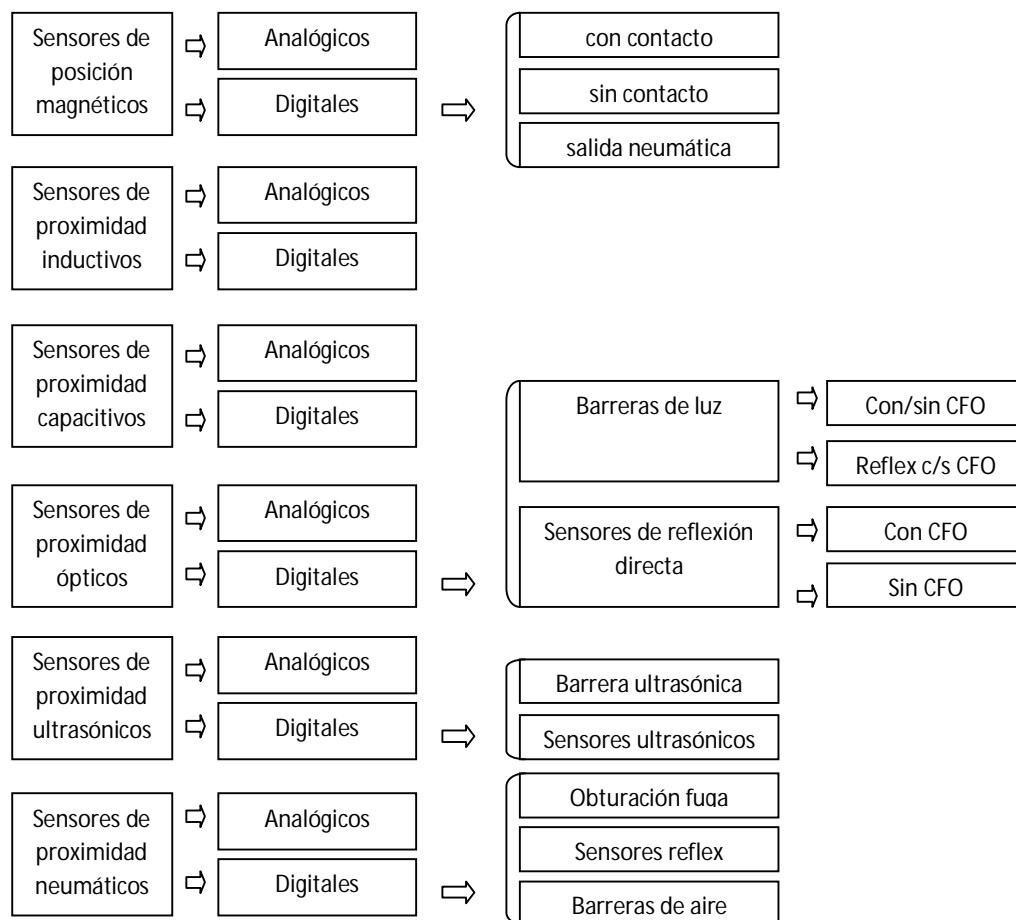
Sensor digital. Se trata de un sensor que convierte una magnitud física en una señal binaria, principalmente en una señal eléctrica con los estados 'ON' 'OFF'.

Sensor analógico. Se trata de un sensor que convierte una magnitud física en una señal eléctrica de tensión o de intensidad.

Los sensores pueden tener varios tipos de salida: Interrupción, trenes de pulsos o salida analógica. En nuestro sistema utilizaremos sensores de proximidad con señal de salida analógica ya que los sensores con este tipo de salida permiten una gestión muy sencilla a través de microcontroladores.

Utilizaremos sensores de proximidad sin contacto, ya que nuestro objetivo es evitar colisiones indeseadas para prevenir daños a personas, maquinas y objetos. Estos sensores permiten una detección precisa y automática de objetos. Tiene unas características de conmutación rápidas y dado que la señal de salida se genera electrónicamente los sensores están libres de rebotes y no crean errores en las señales de salida. Presentan una gran resistencia al desgaste porque no tienen partes móviles.

A continuación podemos ver un esquema distintos tipos de sensores de proximidad:



SENSORES DE POSICION MAGNETICOS.

Existen varios tipos de sensores de proximidad que se incluyen en este grupo, todos ellos con el aspecto común de que utilizan alguna característica magnética para la detección de proximidad o presencia.

SENSOR DE PROXIMIDAD REED.

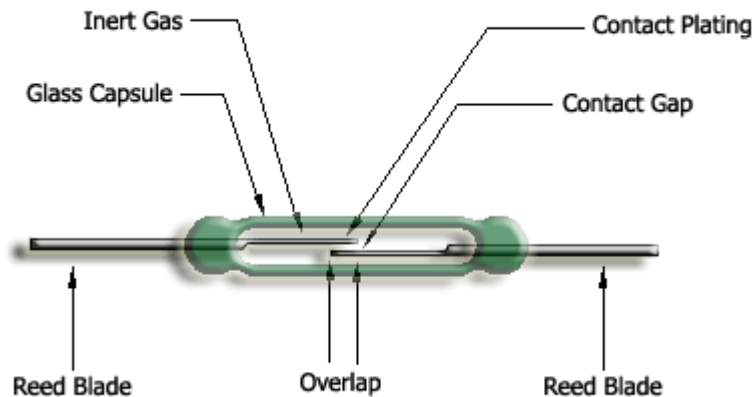


Figura3. Esquema sensor reed

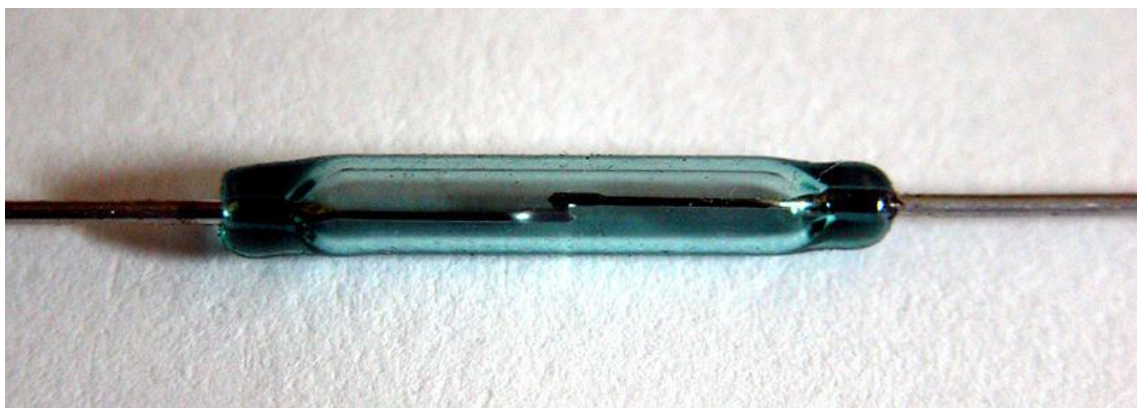


Figura4. Sensor reed KSK-1A52-2025

Su funcionamiento se basa en unas láminas metálicas que entran en contacto cuando un campo magnético se acerca al sensor.

Los reed son utilizados ampliamente en el mundo moderno como partes de circuitos eléctricos. Un uso muy extendido se puede encontrar en los sensores de las puertas y ventanas de las alarmas antirrobo, el imán va unido a la puerta y el reed al marco. En los sensores de velocidad de las bicicletas el imán está en uno de los radios de la rueda, mientras que el reed va colocado en la horquilla.

Algunos teclados de computadoras son diseñados colocando imanes en cada una de las teclas y los reed en el fondo de la placa, cuando una tecla es presionada el imán se acerca y activa su reed. Actualmente esta solución es obsoleta, usándose interruptores capacitivos que varían la condición de un circuito resonante.

SENSOR MAGNETO-INDUCTIVO.

Estos sensores de proximidad, similares a los sensores de proximidad inductivos, tiene un oscilador incorporado (Circuito oscilador LC). Sin embargo, en contraste con los sensores de proximidad inductivos, la bobina osciladora no es del tipo con núcleo de media capa, sino una bobina con un núcleo de capa cerrada, es decir, una bobina con un núcleo de ferrita apantallado. Al acercar un imán permanente, el material del núcleo del oscilador se satura causando con ello una variación en la corriente del oscilador del sensor de proximidad. Una etapa de disparo evalúa el cambio y lo convierte en una señal de salida definida. El rango de conmutación de este tipo de sensores de proximidad depende de la orientación el eje polar del imán. Estos sensores de proximidad solamente reaccionan ante campos magnéticos, y no ante cualquier objeto metálico.



Figura5. Sensor magnetoinductivo RD4946-01

SENSOR EFECTO HALL.

En el mercado existe gran cantidad de sensores industriales para diversos usos, basados en el efecto que descubrió el científico Edwin Herbert Hall. El nombre de Hall, físico norteamericano, ha pasado a la posteridad debido a una singularidad electromagnética que descubrió por causalidad en el curso de un montaje eléctrico: el "Efecto Hall".

Cuando por una placa metálica circula una corriente eléctrica y ésta se halla situada en un campo magnético perpendicular a la dirección de la corriente, se desarrolla en la placa un campo eléctrico transversal, es decir, perpendicular al sentido de la corriente. Este campo, denominado Campo de Hall, es la resultante de fuerzas ejercidas por el

campo magnético sobre las partículas de la corriente eléctrica, sean positivas o negativas.

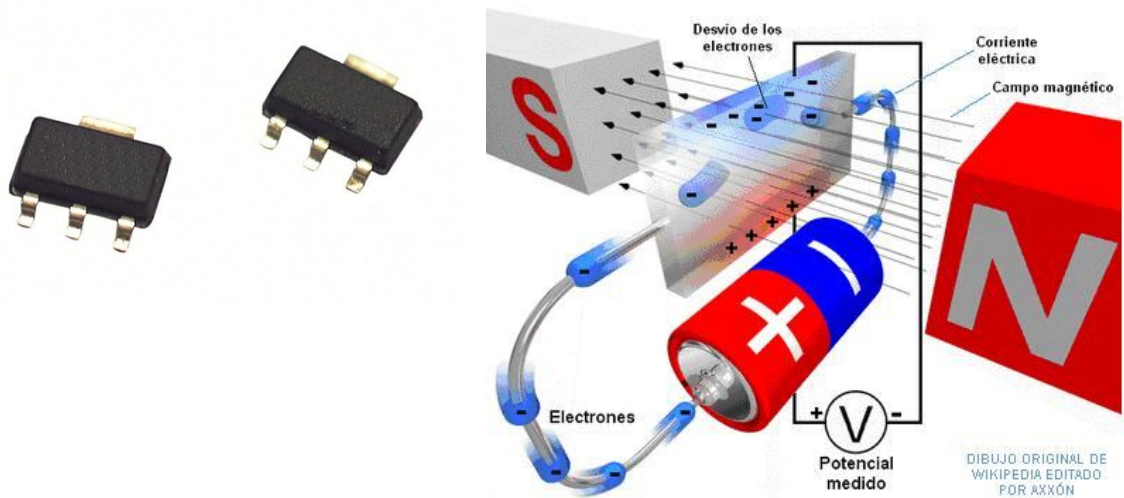


Figura6. Sensor efecto hall SS59ET y su principio de funcionamiento

Si fluye corriente por un sensor Hall y se aproxima a un campo magnético que fluye en dirección vertical al sensor, entonces el sensor crea un voltaje saliente proporcional al producto de la fuerza del campo magnético y de la corriente.

Los sensores basados en efecto Hall suelen constar de un elemento conductor o semiconductor y un imán. Cuando un objeto ferromagnético se aproxima al sensor, el campo que provoca el imán en el elemento se debilita. Así se puede determinar la proximidad de un objeto, siempre que sea ferromagnético.

Una de las aplicaciones de los sensores de efecto Hall que más se ha instalado en la industria, en especial en la automotriz, es como reemplazo del sensor inductivo (basado en un imán permanente y una bobina). Dado que en este caso el sensor, por estar implementado por un semiconductor, tiene la capacidad de poseer electrónica integrada, la señal que sale de los sensores por efecto Hall para uso como detectores de proximidad por lo general ya está amplificada y condicionada, de modo que su utilización es mucho más directa, fácil y económica.

Otra aplicación es la medición de la corriente que circula por un conductor, con lo que se pueden implementar medidores de seguridad sin necesidad de insertarlos en el circuito eléctrico de un sistema donde se maneja potencia. Los sensores pueden estar contruidos en una cápsula de tipo circuito integrado o una de transistor.

SENSOR MAGNETORRESISTIVO.

Este tipo de sensor varía el valor de una resistencia en presencia de un campo magnético externo, como se puede observar en la figura 7.



Figura7. Esquema sensor magnetorresistivo



Figura8. Sensor magnetorresistivo FVD-L

Son un excelente medio para medir tanto desplazamiento angulares como lineales en condición ambientales extrema debido a su alta sensibilidad, ya que un pequeño movimiento en equipos de ruedas, engranajes, levas, barras metálicas, etc, provocan cambios significativos del campo magnético. Otras aplicaciones son la medición de velocidad de rotación y de intensidad de corriente.

También se utilizan en automoción para medir la posición del chasis, del acelerador, posición de los pedales, en la rueda como sensor de velocidad para el ABS. Otros ejemplos lo incluyen en instrumentación y equipos de control que requieren sensores de posición capaces de detectar desplazamientos de decimas de milímetro y en sistemas de encendido electrónico donde es necesario detectar con gran precisión la posición de motores internos.

Por último, y debido a su capacidad para reaccionar ante campos magnéticos muy débiles, por su gran sensibilidad, su utiliza en brújulas electrónicas.

SENSOR WIEGAND.

El efecto Wiegand consiste en la conmutación de la magnetización de la zona central de un hilo ferromagnético cuando se le somete a un campo magnético externo intenso.

El fenómeno fue patentado por J.R. Wiegand en 1981 y si se somete el hilo a un campo magnético longitudinal cíclico, el núcleo y la cubierta experimentan conmutaciones magnéticas (inversiones de polaridad) para intensidades de campo distintas. Los cambios de flujo magnético inducen una tensión en la bobina externa. En la figura9 podemos observar los distintos tipos de sensores wiegand.

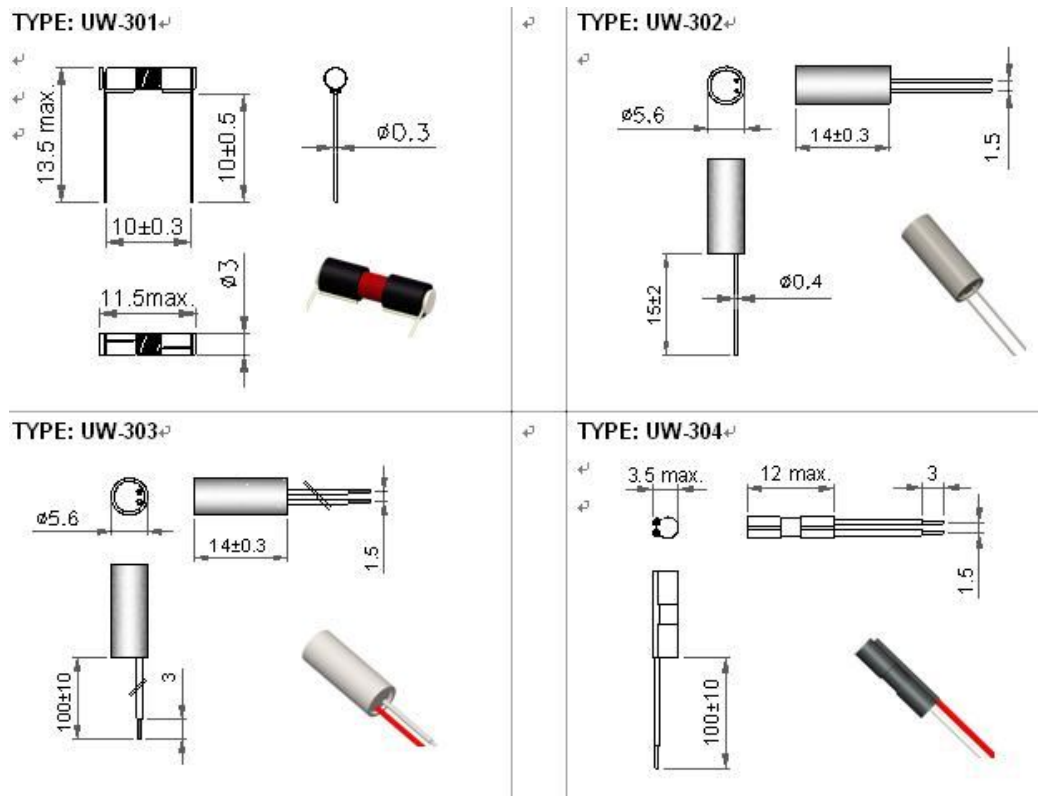


Figura9. Modelos sensor Wiegand UW301, UW302, UW303, UW304

Inicialmente se garantiza que la cubierta y el núcleo estén magnetizados a saturación en direcciones opuestas. Las líneas de flujo de la cubierta circulan por el núcleo y lo mantienen magnetizado. El campo magnético resultante es casi nulo. Si se aplica un campo magnético paralelo al de la cubierta, por ejemplo mediante un imán externo, cuando se alcanza un valor adecuado se produce un cambio brusco de la polaridad del núcleo porque sus dominios magnéticos se reorientan. Las líneas de campo circulan ahora por el exterior del hilo y pueden inducir una tensión en la bobina. Al final, el núcleo y la cubierta quedan magnetizados en la misma dirección.

Los sensores basados en el efecto Wiegand se aplican a la detección sin contactos de campos magnéticos y magnitudes que los pueden hacer variar, como son los movimientos y la posición. Se utilizan, por ejemplo, para detectar la posición del cigüeñal en automóviles, para teclados, sin rebotes, donde el elemento móvil es un hilo Wiegand y el cabezal detector consiste en una bobina y un imán permanente. En tacómetros permiten medir velocidades de rotación pequeñas. En sistemas antirrobo y tarjetas de identificación donde, por ejemplo, medir velocidades de rotación pequeñas. En sistemas antirrobo y tarjetas de identificación donde, por ejemplo, cada bit corresponde a un hilo Wiegand incrustado dentro de la tarjeta. Se han aplicado también a la medida del flujo de un fluido, empleando como sensor primario una turbina con paletas no magnética inmersa en el seno del fluido que gira debido al flujo

y lleva un hilo Wiegand incrustado. El cabezal detector incorpora una bobina y dos imanes y pueden situarse fuera de la tubería.

Los sensores de proximidad magnéticos no son los más adecuados para nuestra aplicación ya que solo reaccionan ante campos magnéticos y por tanto deberíamos dotar de un imán a cada objeto que deseemos detectar, algo impensable si deseamos detectar objetos de diversos tipos y tanto en entorno cerrados como abiertos. A parte de dotar cada objeto con imán existe la posibilidad de que el objeto no sea detectado debido a que el rango de conmutación depende de la orientación del imán. Además un sistema multisensor, como tenemos pensado para nuestro sistema anticollisión, es difícil de conseguir con este tipo de sensores ya que debe haber una distancia mínima entre ellos para que no se produzcan conmutaciones indeseadas. También existe el problema de que al trabajar con materiales ferromagnéticos hay una histéresis que no podemos despreciar y que en ocasiones podrían generar más conmutaciones erróneas.

SENSORES DE PROXIMIDAD INDUCTIVOS.

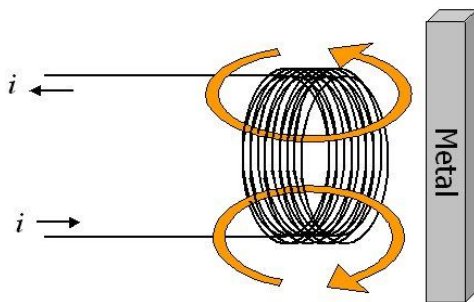


Figura10. Esquema sensor inductivo



Figura11. Tipos de sensores inductivos LBSP142

En la figura 10 podemos observar el principio de funcionamiento de los sensores inductivos y en la figura 11 observamos diferentes tipos de estos sensores

Los componentes más importantes de un sensor de proximidad inductivo son un oscilador (circuito resonante LC), un rectificador demodulador, un amplificador biestable y una etapa de salida. Estos componentes los podemos observar la figura 12

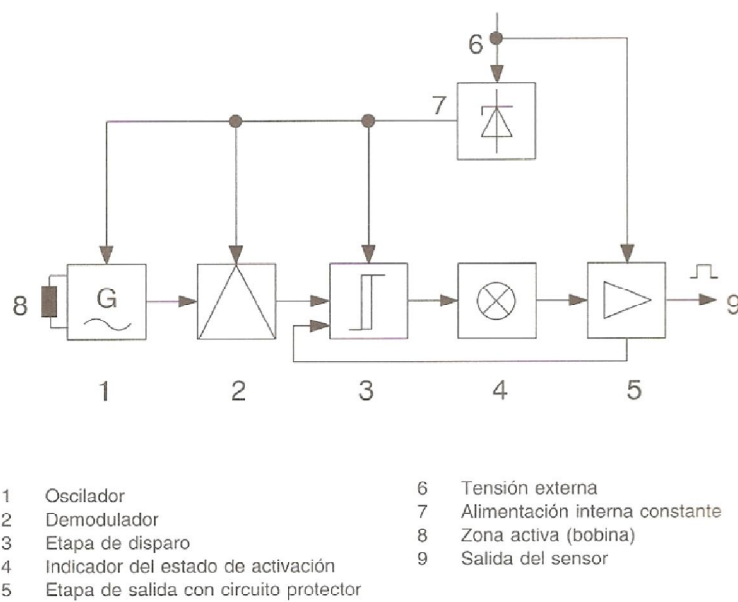


Figura 12. Esquema eléctrico sensor inductivo

El campo magnético, que es dirigido hacia el exterior, es generado por medio del núcleo de ferrita de una bobina osciladora y de un apantallado adicional. Esto crea un área limitada a lo largo de la superficie activa del sensor de proximidad inductivo, la cual se conoce como zona activa de conmutación.

Quando se aplica una tensión al sensor, el oscilador se activa y fluye una corriente de reposo definida. Si un objeto conductor de electricidad se introduce en la zona activa de conmutación, se crean unas corrientes parasitas que restan energía al oscilador. La oscilación se atenúa y esto produce un cambio en el consumo de corriente del sensor de proximidad. Los dos estados: oscilación atenuada y oscilación si atenuar, se evalúan electrónicamente.

Existen varias aplicaciones para este tipo de sensores. Son utilizados para la detección de la posición del émbolo en un cilindro neumático o hidráulica, como la que podemos observar en la figura 13 pero siempre para objetos metálicos. Otra aplicación es la detección de transportadores metálicos de piezas en una cinta, así como la detección de la posición de un árbol de levas. También se utilizan para la medición de velocidad y sentido de rotación.

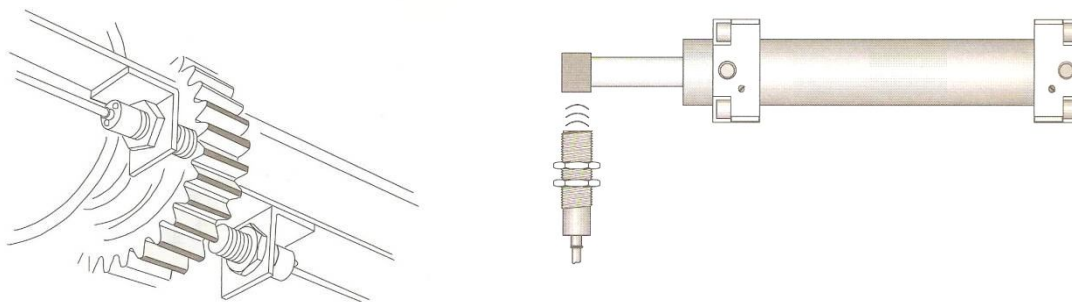


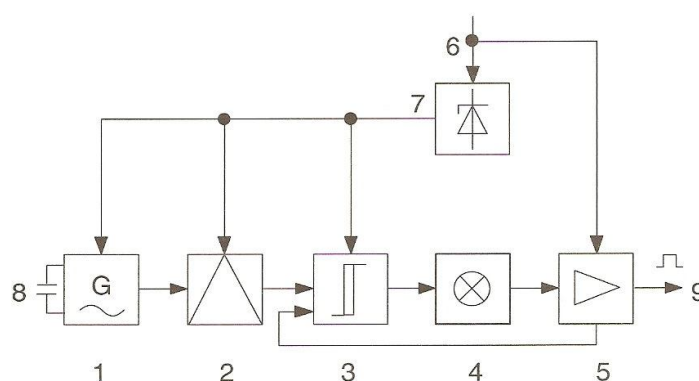
Figura 13. Aplicación sensor inductivo como final de carrera

Estos sensores de proximidad no pueden ser utilizados en nuestra aplicación, aunque cumplen con características como: tensión de funcionamiento, distancia de detección, potencia consumida e incluso tamaño. Sin embargo solo sirven para detectar materiales y objetos conductores de electricidad.

SENSORES DE PROXIMIDAD CAPACITIVOS.

El principio de funcionamiento de un sensor de proximidad capacitivo, está basado en la medición de los cambios de capacitancia eléctrica de un condensador en un circuito resonante RC, como muestra la figura 14, ante la aproximación de cualquier material.

En un sensor de proximidad capacitivo, entre un electrodo "activo" y uno puesto a tierra, se crea un campo electrostático disperso. Generalmente también se halla presente un tercer electrodo para compensación de las influencias que pueda ocasionar la humedad en el sensor de proximidad.



- | | | | |
|---|---|---|--------------------------------|
| 1 | Oscilador | 6 | Tensión externa |
| 2 | Demodulador | 7 | Alimentación interna constante |
| 3 | Etapas de disparo | 8 | Zona activa (condensador) |
| 4 | Indicador del estado de activación | 9 | Salida del sensor |
| 5 | Etapas de salida con circuito protector | | |

Figura 14. Esquema eléctrico sensor capacitivo

En la figura 15 y 16 podemos observar tanto las distintas partes que componen un sensor capacitivo como los distintos tipos que existen en el mercado

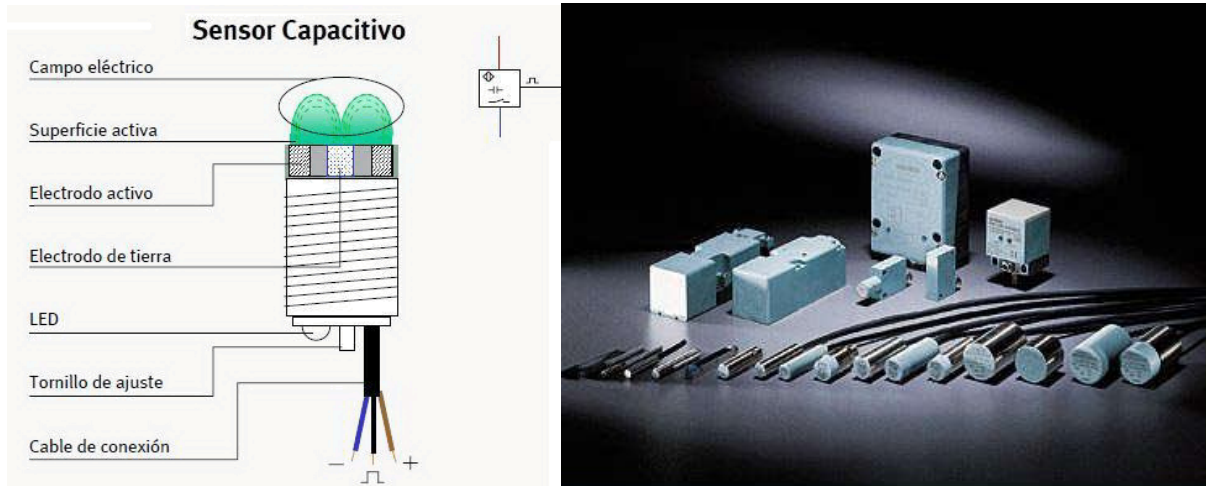


Figura 15. Partes de un sensor capacitivo **Figura 16. Tipos de sensores capacitivos M12,etc...**

Si un objeto o un medio (metal, plástico, vidrio, madera, agua), irrumpe en la zona activa de conmutación. La capacitancia del circuito resonante se altera.

Este cambio en la capacitancia depende esencialmente de los siguientes parámetros: la distancia entre el medio y la superficie activa, las dimensiones del medio y su constante dieléctrica.

La sensibilidad de la distancia de detección de la mayoría de los sensores de proximidad capacitivos puede ajustarse por medio de un potenciómetro. De esta forma es posible eliminar la detección de ciertos medios. Por ejemplo, es posible determinar el nivel de un líquido a través de la pared de vidrio de su recipiente.

Debido a esta propiedad de reaccionar ante una amplia gama de materiales, el sensor de proximidad capacitivo es más universal en aplicaciones que el sensor de proximidad inductivo. Por otro lado, los sensores de proximidad capacitivos son sensibles a los efectos de la humedad en la zona activa de detección. Muchos fabricantes, por ejemplo, utilizan un electrodo auxiliar para compensar los efectos de la humedad, rocío o hielo, reduciendo así estas perturbaciones.

Hay campos de aplicación particulares donde la utilización de sensores de proximidad capacitivos proporcionan diferentes ventajas. Sin embargo en la detección de objetos no metálicos, también compiten como alternativa viable los sensores de proximidad ópticos.

También por razones de coste, en la detección de objetos metálicos se prefieren generalmente los sensores de proximidad inductivos a los capacitivos.

Los sensores de proximidad capacitivos son adecuados, por ejemplo, para supervisar los niveles de llenado en contenedores de almacenamiento. Otras áreas de aplicación incluyen la detección de materiales no metálicos.

Los objetos de goma, cuero, plástico y otros materiales, son difíciles de detectar por sensores ópticos de reflexión directa y, en según qué aplicaciones, la utilización de sensores ultrasónicos puede resultar excesivamente costosa.

Este tipo de sensores son muy utilizados para detectar niveles de líquidos, como podemos observar en la figura 17, a través de paredes finas de recipientes de plástico, tubos de vidrio de inspección, etc. El grueso de la pared debe limitarse de tal forma que permita al sensor de proximidad capacitivo reaccionar solo con el propio contenido del recipiente.

Pero no solo son utilizados para detectar niveles de líquidos, si no que debido a su gran capacidad para detectar diferentes materiales, también son adecuados para la detección de materiales pulverulentos o granulados en contenedores o silos

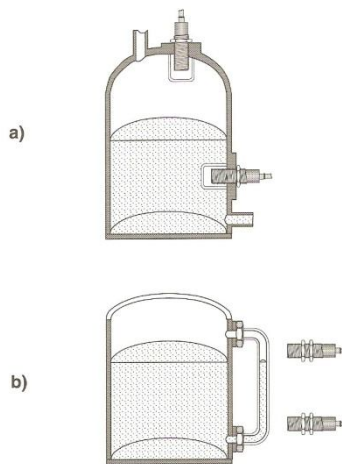


Figura 17. Aplicación sensores capacitivos como detector de nivel

Con los sensores de proximidad capacitivo es posible, por ejemplo, verificar el volumen de llenado de contenedores de productos alimentados a través de cajas selladas, como se puede observar en la figura 18

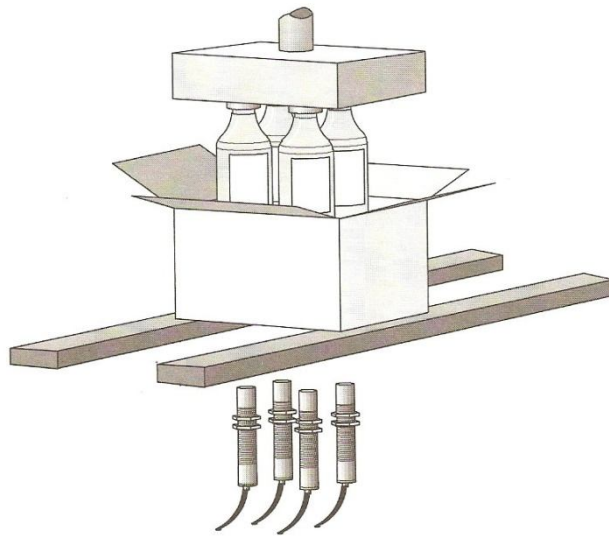


Figura 18. Aplicación sensor capacitivo como detector de presencia

Los sensores de proximidad capacitivos reaccionan al cobre que contienen los hilos o cables eléctricos de diámetro relativamente pequeño, donde los sensores de proximidad inductivos no reaccionan o lo hacen a distancias inferiores, y por tanto se pueden emplear para la supervisión del bobinado de cables e hilos eléctricos.

Parece que este tipo de sensores podrían utilizarse en nuestra aplicación, sin embargo existe el problema del tamaño ya que estos sensores son sensores de unos 30mm de diámetro que impiden que sean montados en un brazo robot debido a la superficie de la que disponemos para colocar un sistema multisensor.

SENSORES DE PROXIMIDAD ULTRASONICOS.



Figura 19. Sensor ultrasónico SRF05

El funcionamiento está basado en la emisión y reflexión de ondas acústicas, entre un emisor, un objeto y un receptor, evaluando el tiempo que tarda en desplazarse el sonido.

Un sensor de ultrasonido puede dividirse en tres módulos principales, el traductor ultrasónico, la unidad de evaluación y la etapa de salida, como muestra la figura 19.

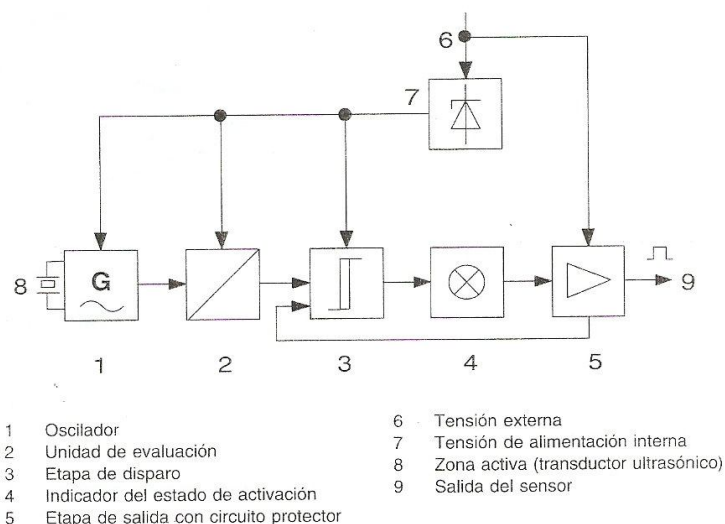


Figura 20. Esquema eléctrico sensor ultrasónico

El transmisor ultrasónico emite ondas de sonido en un rango inaudible, general mente entre 30 y 300 KHz. En muchos casos el transmisor cambia de emisor a receptor. Los filtros internos del sensor comprueban si el sonido recibido es realmente el eco de las ondas emitidas.

La velocidad de los sensores de ultrasonido está limitada por la máxima frecuencia de repetición de pulsos, que puede oscilar entre 1 y 25 Hz.

La principal ventaja de los sensores de proximidad ultrasónicos reside en el hecho que pueden detectar una amplia gama de diferentes materiales. La detección es independiente de la forma, color y material, mientras que el material puede ser sólido, fluido o en forma de polvo. La detección no se ve afectada por la suciedad, ni por las atmósferas con vapores o humos.

Los sensores de ultrasonido están generalmente disponibles en forma de sensores de reflexión directa, donde emisor y receptor se hallan en un mismo cuerpo, aunque por otro lado se dispone de barreras ultrasónicas que tienen el emisor y el receptor en cuerpos separados.

Algunas de las ventajas de los sensores ultrasónicos son un rango de detección amplio (hasta varios metros), detección del objeto independientemente de su forma color o material, detección de objetos transparentes, insensibilidad a la suciedad y al polvo, posibilidad de aplicación al aire libre y por supuesto que la detección se realiza sin contacto.

Por el contrario el uso de ultrasonidos también tiene algunos inconvenientes como la dificultad para detectar objetos con superficies inclinadas, ya que estas desvían las ondas de sonido, o materiales como las telas que absorben el sonido. Otra desventaja es la lentitud con la que reaccionan, ya que la frecuencia de conmutación máxima está entre 1 y 125 Hz. Y además son más caros que los sensores ópticos.

Generalmente los sensores de ultrasonido están equipados con un potenciómetro para el ajuste del rango de funcionamiento. También hay ejecuciones con dos potenciómetros para ajustar una ventana de conmutación, así como ejecuciones especiales programables, con las que pueden seleccionarse diferentes rangos de funcionamiento por medio de un interface electrónico.

Algunos están equipados con entradas sincronizadas, con lo que es posible un funcionamiento libre de interferencias, cuando se utiliza un sistema multisensor que requiere sensores próximos entre sí.

Los sensores de proximidad ultrasónicos se utilizan para supervisar los niveles de llenado de silos.

También han demostrado su fiabilidad para el control de transportadores automáticos en almacenes.

Otras aplicaciones adicionales pueden ser el control de bucle entre rodillos en una cinta transportadora, como se puede observar en la figura 21, clasificación de objetos según las diferentes alturas o detección de grueso en lotes.

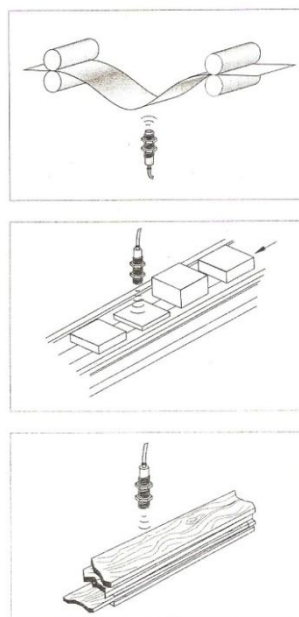


Figura 21. Aplicaciones sensor de ultrasonido. Detección de bucle, clasificación por alturas y detección de grueso

Con este tipo de sensores se pueden crear sistemas anticolidión para asistencia en la conducción en la que se requiere en esencia detectar objetos cercanos alrededor del vehículo y no una descripción del entorno.

La utilización de ultrasonido se considera como la técnica más adecuada en este tipo de sonorización, sus características de operación son superiores frente a otros métodos, presentan haces amplios que reducen las necesidades de barridos y buena sensibilidad en distancias dentro del rango de 20 a 100 cm.

Los sensores de ultrasonido son colocados en la periferia del vehículo para ayudar a la navegación de la silla e impedir colisiones bruscas con obstáculos.

El objeto a detectar debe tener un tamaño mínimo, ya que si las ondas de ultrasonido se dirigen a un objeto demasiado pequeño, cualquier objeto que se halle al lado o de tras puede interferir.

Para la detección de objetos con este tipo de sensor se requiere una distancia mínima de detección, ya que requiere un mínimo tiempo de proceso para detectar el eco del ultrasonido, no pueden funcionar si una cierta zona muerta. En distancias cortas los lóbulos secundarios del cono ultrasónico pueden producir pulsos de error.

SENSORES DE PROXIMIDAD OPTICOS.



Figura 22. Sensor infrarrojo GP2D12J0000F

Los sensores de proximidad ópticos utilizan medios ópticos y electrónicos para la detección de objetos. Para ello se utiliza luz roja o infrarroja. Los diodos semiconductores emisores de luz (LEDs) son una fuente particularmente fiable de luz roja e infrarroja. Son pequeños y robustos, tienen una larga vida útil y puede modularse fácilmente. Los fotodiodos y fototransistores se utilizan como elementos receptores. Cuando se ajusta un sensor de proximidad óptico la luz roja tiene la ventaja frente a la infrarroja de que es visible.

La luz infrarroja (invisible) se utiliza en ocasiones en las que se requieren mayores prestaciones, por ejemplo, para cubrir mayores distancias. Además, la luz infrarroja es menos susceptible a las interferencias (luz ambiental).

Con ambos tipos de sensores de proximidad ópticos, la supresión adicional de las influencias de luz externas se alcanza por medio de la modulación de la señal óptica. El receptor (con la excepción de los sensores de barrera) se sintoniza con los pulsos del

emisor. Con sensores de barrera se utiliza un pasa banda eléctrico en el receptor. Particularmente en el caso de luz infrarroja, la utilización de filtros de luz diurna, mejora aún más, su insensibilidad a la luz ambiental.

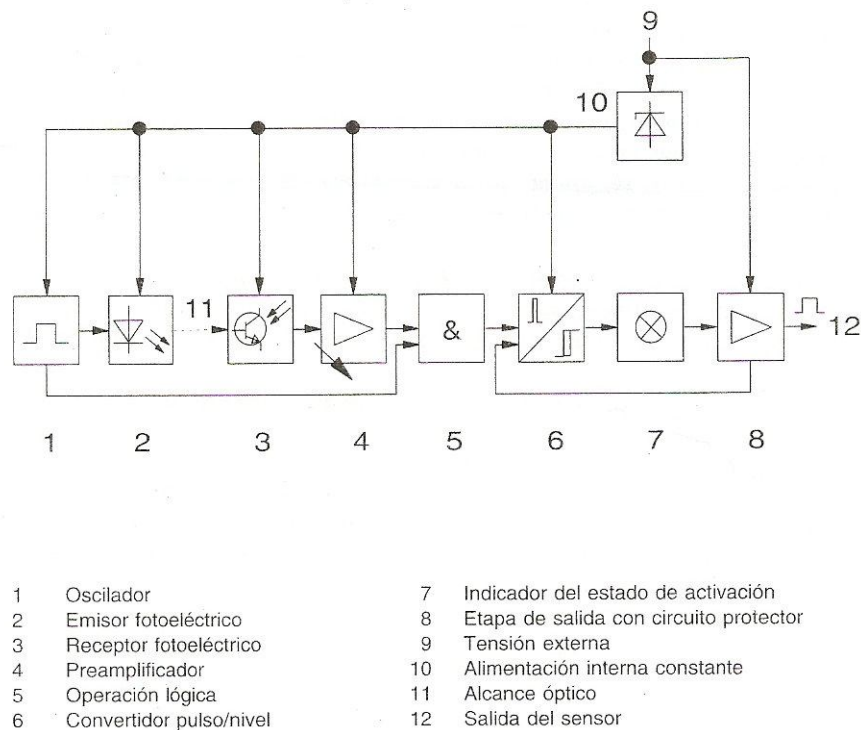


Figura 23. Esquema de un sensor de proximidad óptico (emisor y receptor en el mismo cuerpo)

Los sensores de proximidad ópticos generalmente incorporan medidas de protección:

- Protección contra la polaridad inversa
- Protección de las salidas contra cortocircuito
- Protección contra picos de tensión.

Los sensores de proximidad ópticos consisten básicamente en dos partes: el emisor y el receptor. Dependiendo del tipo y de la aplicación, se requiere adicionalmente reflectores y cables de fibra óptica.

El emisor y el receptor se hallan instalados en un cuerpo común (sensores de reflexión directa o retrorreflexión), o en cuerpos separados (sensores de barrera).

Con sensores de barrera y de retrorreflexión, se distinguen las siguientes funciones de conmutación:

- Método de detección por luz

La salida conecta cuando el rayo de luz no está interrumpido por un objeto (contactos normalmente abiertos). En el caso de una barrera que se active por luz, la salida en el receptor está activada si no hay ningún objeto que obstruya el rayo de luz.

- Método de detección por oscuridad

La salida está inactiva (sin conmutar) cuando el rayo de luz no está interrumpido por ningún objeto (contactos normalmente cerrados). En el caso de una barrera que se active por oscuridad, la salida del receptor se conecta si hay un objeto que irrumpe en el rayo de luz.

Las funciones de conmutación de los sensores de reflexión directa son como sigue:

- Método de detección por luz

La salida cierra si el objeto a detectar se introduce en el rayo de luz (Salida normalmente abierta).

- Método de detección por oscuridad

La salida abre si el objeto a detectar se introduce en el rayo de luz (Salida normalmente cerrada).

El emisor aloja la fuente de emisión de luz roja o infrarroja, la cual, y según las leyes de la óptica, se propaga en línea recta y puede ser desviada, enfocada, interrumpida, reflejada y dirigida. Esta luz es aceptada por el receptor, separada de la luz externa, y evaluada electrónicamente.

El sensor de proximidad se monta con un apantallamiento interno, que es aislado del cuerpo. Los componentes electrónicos son encapsulados y se dispone un potenciómetro externo en el lado de la salida del cable, para ajustar la sensibilidad.

Los sensores de proximidad ópticos a menudo están expuestos a la contaminación por polvo, virutas o lubricantes durante su funcionamiento. Esta contaminación puede ser causa de interferencias en los sensores de proximidad ópticos. Tanto el ensuciamiento de las lentes que forman parte de la óptica del sensor de proximidad como el de los reflectores en los de retrorreflexión, puede ser causa de fallos en el funcionamiento.

Para conseguir un funcionamiento fiable, deberían tomarse las siguientes medidas:

- Hacer funcionar el sensor de proximidad óptico con suficiente margen de operación

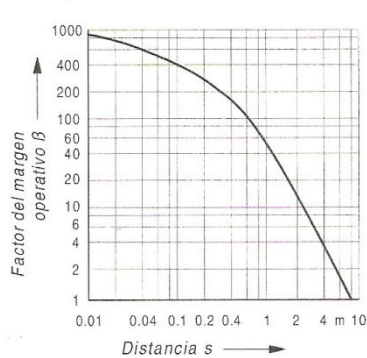
- Realizando ensayos previos
- Seleccionando un sensor de proximidad con suficiente margen de funcionamiento.
- Utilizando sensores de proximidad con ayudas al ajuste, por ejemplo, parpadeo del LED en las zonas limite de detección.
- Utilizando sensores de proximidad con indicación automática de ensuciamiento.

Los sensores de proximidad ópticos tienen un cierto margen de funcionamiento (también conocido como reserva funcional) β , que es el cociente de la potencia real de la señal óptica en la entrada del receptor P_R dividida por la potencia de la señal óptica mínima detectable en el umbral de conmutación P_T :

$$\beta = P_R / P_T$$

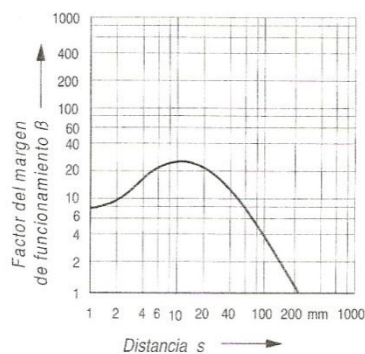
Si la emisión óptica recibida está en el nivel del umbral de conmutación, esto significa $\beta=1$, es decir, no hay margen de funcionamiento. Si el factor es, por ejemplo, $\beta=1,5$ significa que se dispone de un margen de funcionamiento del 50%.

El factor β depende por una parte de la distancia entre el emisor y el receptor en el caso de los sensores de barrera, entre el emisor y el reflector en el caso de los de retrorreflexión o entre el sensor y el objeto en el caso de los sensores de reflexión directa, como muestran las figuras 23, 24 y 25 respectivamente.



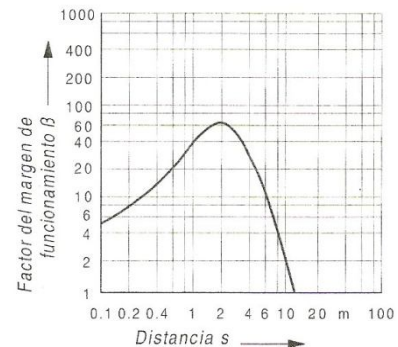
Reserva funcional sensor de barrera

Figura 24



Reserva funcional sensor de retrorreflexión

Figura 25



Reserva funcional reflexión directa

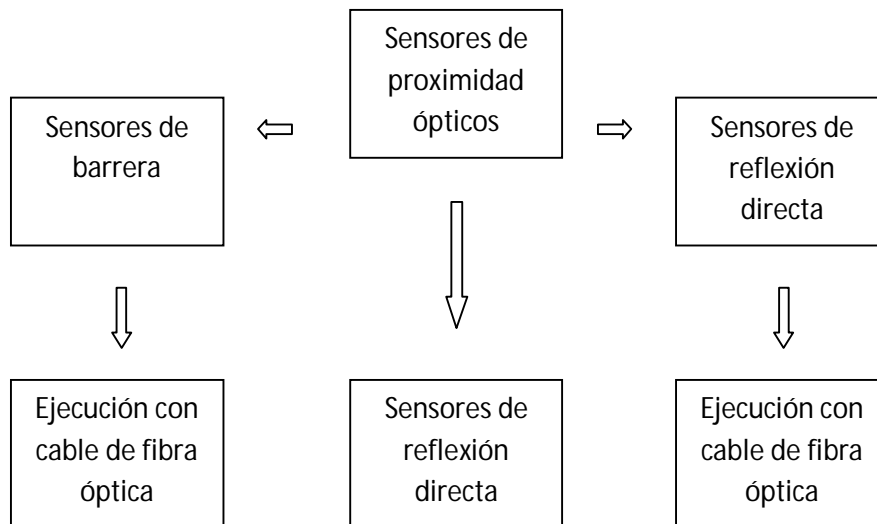
Figura 26

A parte de la contaminación, pueden ser la causa de que el sensor quede fuera del margen de funcionamiento:

- Sobrepasar el margen de seguridad de detección

- Cambios en la superficie del material de los objetos detectados
- Montaje incorrecto
- Envejecimiento del diodo emisor.

Como se puede comprobar, durante el apartado de sensores de proximidad ópticos, estamos ablando de diferentes variantes, las cuales podemos ver en el siguiente esquema



Sin embargo de todas estas variantes es evidente que solo una nos sirve para nuestra aplicación. Los sensores de proximidad ópticos de reflexión directa.

Debido a que el sensor estará montado en un brazo robot y no sabemos donde se encontrara el objeto a detectar debemos eliminar como posibles sensores los sensores ópticos de barrera y los de retrorreflexión, ya que no podemos tener en cuerpos separados a emisor y receptor, ni utilizar reflectores.

SENSORES DE REFLEXION DIRECTA.

En estos sensores el emisor y receptor se hallan alojados en un mismo cuerpo. El objeto a detectar refleja directamente un porcentaje de la luz emitida, como muestra la figura 27, activando con ello el receptor. Dependiendo del diseño del receptor, la salida es activada (función normalmente abierta) o desactivada (función normalmente cerrada). La distancia de detección depende mucho de la reflectividad del objeto. El tamaño superficie, forma, densidad y color del objeto, así como el ángulo de incidencia del rayo, determinan la intensidad de la luz reflejada de forma que, en general, solo pueden detectarse objetos a distancias cortas, del orden de unos pocos decímetros. El fondo debe absorber o desviar la emisión de luz, es decir, cuando no hay objeto el rayo de luz reflejado debe estar netamente por debajo del umbral de respuesta del circuito receptor.



Figura 27. Funcionamiento sensor de reflexión directa

Las características técnicas de estos sensores cumplen las exigencias de nuestra aplicación:

Tensión de alimentación	10.....30Vdc
Alcance	50mm hasta 2m (ajustable)
Intensidad de ruptura	100....500mA
Frecuencia de conmutación	10.....2000Hz.

Ventajas de los sensores de reflexión directa:

- No requiere reflector adicional
- El objeto puede ser reflectante, especular o transparente y hasta traslucido mientras refleje un porcentaje suficientemente elevado del rayo de luz que recibe
- Permiten detectar en posición frontal, es decir en la dirección del rayo de luz
- Dependiendo del ajuste el objeto puede detectarse selectivamente frente a un fondo.

Desventajas de los sensores de reflexión directa:

- La respuesta no es completamente lineal. Por ello, los sensores de reflexión directa no son tan adecuados como los sensores de barrera si se necesita una elevada precisión de respuesta lateral

Existen unos factores de corrección definidos para este tipo de sensores, que se muestran en la tabla1:

Tabla1. Factores de corrección para sensores ópticos

Material	Factor
Cartón, blanco	1.0
Poliestireno expandido, blanco	1.0....1.2
Metal brillante	1.2....2.0
Madera basta	0.4....0.8
Tejido de algodón blanco	0.5....0.8
Cartón, blanco mate	0.1
Cartón, blanco brillante	0.3
PVC, gris	0.4....0.8

La distancia de conmutación debe multiplicarse por el factor de corrección.

La sensibilidad de estos sensores es ajustable. El efecto del sensor de reflexión directa depende de la diferencia entre la reflexión del objeto y la del fondo. Con contrastes muy pequeños, el umbral de respuesta debe ajustarse, si es preciso, modificando la sensibilidad del sensor de proximidad (potenciómetro de 1 vuelta o multivuelta) de forma que el objeto sea detectado con fiabilidad incluso bajo estas circunstancias difíciles.

Sin embargo debe tenerse en cuenta un margen de tolerancia en relación con el envejecimiento, fluctuaciones de la tensión o la temperatura y suciedad. Por esta razón, cuando se realizan los ajustes no debe utilizarse todo el margen de tolerancia.

Cuando se ajusta el sensor de reflexión directa con su potenciómetro, debe dejarse un cierto margen teniendo en cuenta los cambios en las condiciones del objeto, del grado de suciedad del sensor o de la contaminación de la atmósfera. De cerca, los ajustes funcionales raramente pueden producir problemas.

Algunos sensores de reflexión directa tienen un LED intermitente incorporado para facilitar un ajuste correcto, el cual parpadea si el objeto no se detecta claramente.

Existen objetos que podrían generar problemas en la detección.

- Objetos transparentes como vidrio claro, plexiglás claro, lámina transparente, etc. Estos materiales tienen generalmente superficies reflectantes lisas, con lo que permiten la utilización de sensores de reflexión directa.

Condición: La superficie a detectar debe estar alineada perpendicularmente con la dirección del rayo de luz.

- Objetos con baja reflexión, como por ejemplo, plásticos negro mate, goma negra, materiales oscuros con superficies rugosas, tejidos oscuros, acero pulido.

Los sensores de reflexión directa no reaccionan ante este tipo de materiales, o solo lo hacen a distancias muy cortas

Para estos problemas se pueden buscar alternativas como el uso de sensores capacitivos o de ultrasonido para aproximación frontal.

2.2.1 Aplicaciones de los sistemas anticolisión

Se ha realizado un exhaustivo estudio de diferentes proyectos en desarrollo o ya desarrollados con anterioridad que utiliza sistemas anticolisión, con el fin de encontrar la mejor solución posible para el proyecto de fin de carrera que se plantea.

El primer proyecto que queremos destacar es un vehículo móvil autónomo [4], en este caso una silla de ruedas manejada por el usuario mediante voz, cuyo esquema podemos observar en la figura 28. Como cualquier vehículo autónomo necesita obtener información del entorno que le rodea para poder desenvolverse en él. Esta información la puede obtener de distintos tipos de sistemas sensoriales. La elección de unos tipos de sensores u otros, está condicionado por los requerimientos técnicos de la aplicación. El sistema sensorial que presenta este proyecto está formado mediante transductores ultrasónicos e infrarrojos.

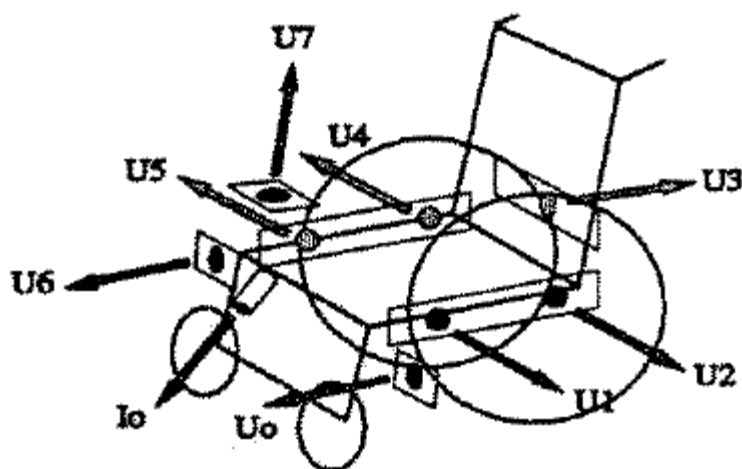


Figura 28. Dibujo silla de ruedas con sistema anticolisión

Para obtener información del entorno, el sistema sensorial realiza medidas de la distancia libre desde el vehículo a los objetos que puedan existir alrededor. Para un correcto funcionamiento estas medidas las realiza mediante técnicas sin contacto, se aprovecha para ello la reflexión de una onda ultrasónica o luz infrarroja, según el caso, lanzada desde uno de los puntos que se pueden ver en el esquema anterior. Para el caso de los ultrasonidos se determinara la distancia a partir del tiempo de ida y vuelta al obstáculo y para el caso de los infrarrojos utilizan la triangulación, ya que el emisor y receptor están separados una distancia conocida.

El sistema realiza medida solo en determinadas direcciones, las que más interesen en función de los movimientos que se estén llevando a cabo o que se vayan a realizar.

No realiza las medidas con una precisión y resolución elevada ya que no se pretende un riguroso posicionamiento.

Para el sistema sensorial utiliza sensores de ultrasonido e infrarrojos ya que son sensores de bajo coste que permite la detección de objetos sin contacto.

Con este sistema sensorial, que se ha dispuesto en anillo, se pretende que el móvil reaccione de forma inmediata ante la presencia de obstáculos en su entorno, sobre todo en las direcciones en que, por su trayectoria, pudiese colisionar con los mismos. También permitirá la detección, mediante sensores infrarrojos, de irregularidades sobre la que se desplaza, sobre todo para la detección de escalones para evitar una posible caída.

Otro proyecto a destacar es el DISEÑO Y CONSTRUCCION DE UN DISPOSITIVO DE ALERTA DE OBSTACULOS [5].

Este proyecto desarrolla un sistema de detección de obstáculos para personas ciegas. Está compuesto por un emisor y receptor ultrasónico, manejado por un microcontrolador. Un bloque emisor genera ondas ultrasónicas, que al chocar con el obstáculo se reflejan y son captadas por el emisor. Tras la elección de un umbral en el microcontrolador, este informa la presencia o no de un objeto activando un motor, que produce una vibración, ubicado en la cintura del usuario. En la figura 29 se muestra el sistema DAO en uso por un usuario ciego.



Figura 29. Sistema DAO en uso por un invidente

El sistema consta de dos dispositivos: un transmisor, encargado de enviar una onda ultrasónica con amplitud y frecuencia controlada, y un receptor. La frecuencia de

trabajo es igual o superior a 40Khz y la amplitud de 2 a 3V, pues es un sistema que funciona con baterías. El receptor se encarga de recibir la onda ultrasónica y entregar una salida de voltaje determinado en función de la distancia.

Se puede ampliar el rango de alcance del dispositivo mediante un circuito amplificador que permita amplificar el tren de pulsos y así detectar objetos a mayor distancia.

Dado que la señal captada por el receptor es débil es necesaria una etapa de amplificación.

El dispositivo es controlado por un microcontrolador que se encarga de procesar la información de los bloque de amplificación y recepción. El programa está diseñado para enviar una señal a los actuadores cuando un obstáculo esta cerca.

El microcontrolador genera ondas cuadradas que son enviadas hacia el transmisor para que este emita la señal. Es una de las partes más importantes ya que de esto depende la señal que enviara el transmisor y que captara el receptor. Este bloque es el encargado de excitar los transmisores y enviar la señal de respuesta adecuada a los actuadores.

El sistema está dotado de unos actuadores que son dos pequeños motores que vibran cuando el obstáculo se encuentra cerca.

Por último un proyecto para mejorar el agarre usando sensores ópticos de proximidad, mostrado en la figura 30.



Figura 30. Tres dedos con sensores ópticos en su cara interna

Se trata de un sistema que ubica sensores de proximidad ópticos en la que permitan captar un punto de agarre inicial sin necesidad de establecer un contacto previo [6]. Los sensores están ubicados en la parte interna de los dedos del robot y el sistema es utilizado como complemento de las técnicas de agarre ya existentes.

Es importante en un robot la capacidad de manipular objetos con fiabilidad y hasta ahora la mayoría de métodos para el agarre de objetos está basado en sensores de largo alcance. Muchos robots han tenido gran éxito con los sistemas de largo alcance,

sin embargo variaciones en el objeto o ubicaciones reducen el éxito de un agarre correcto y se ha detectado que el 65% de los errores producidos a la hora del agarre es debido a la utilización de estos sensores de largo alcance.

El propósito de los sensores ópticos es proporcionar información de las superficies cercanas a los dedos. Los sensores ópticos constan de un emisor y un fotoreceptor. La luz es reflejada por la superficie y recibida por el fotoreceptor. La amplitud e intensidad de la señal recibida varía según distancia, orientación y características de la superficie a detectar. El tamaño del sensor es un dato a tener en cuenta debido a su ubicación y es necesario que sean sensores de tamaño reducido.

Otro dato importante sobre este proyecto es la correcta distribución y ubicación de los sensores ópticos. La configuración consta de tres sensores dispuesto e un triangulo en la cara del dedo para estimar posición y un cuarto sensor en ángulo de 45° para ampliar campo de visión.

El sistema completo de detección lo forman doce sensores, cuatro en cada dedo, controlados por un microcontrolador situado en la muñeca del robot que realiza 16 lecturas por ciclo. Los datos son recogidos y luego se muestran por un convertidor A/D de 10 bits.

Realizando varias medidas con los sensores se pretende estimar la distancia y orientación de la superficie que se encuentra delante del dedo. El inconveniente de trabajar con sensores ópticos es que la respuesta variara en función de la reflectividad, difusividad, etc... de la superficie a detectar.

2.3 Protocolo CANbus

CANbus es un protocolo de comunicación serie para el intercambio de información entre unidades de control electrónico que soporta un eficiente control en Tiempo Real con un alto nivel de fiabilidad. Fue desarrollado para la industria de la automoción y está estandarizado desde 1993 por la norma ISO 11898-1 [7].

La siglas CAN significa Controller Area Network (Red de controladores locales) y Bus se entiende como un elemento de comunicaciones que permite transportar una gran cantidad de información entre las unidades de control abonadas a un sistema.

CANbus permite aumentar las funciones presentes en un sistema de control sin modificar los nodos ya existentes, además de que estas funciones pueden estar repartidas entre las distintas unidades de control. Permite disminuir notablemente el cableado en cualquier sistema de control, puesto que si una unidad de mando dispone de una información esta puede ser utilizada por el resto de unidades de mando sin que sea necesario que cada una de ellas reciba la información directamente de dicho sensor.

Otra ventaja obvia, además de que las funciones pueden ser repartidas entre distintas unidades de mando, es que incrementar las funciones de las mismas no presupone un coste adicional excesivo.

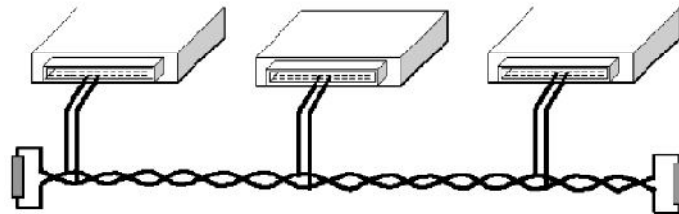


Figura 31. Sistema CANbus

2.3.1 Características del protocolo CANbus

CAN está basado en un modelo MAESTRO-ESCLAVO orientado al mensaje, por lo que la información que se transmite se descompone en tramas a las que se asocia un identificador. Las principales características del CAN son las siguientes:

- Prioridad de mensajes.
- Garantía de tiempos de latencia.
- Configuración flexible.
- Recepción multicast con sincronización de tiempos.
- Sistema de consistencia de datos muy robusto.
- Multimaster.
- Detección y señalización de errores.
- Retransmisión automática de mensajes erróneos.
- Distinción entre fallos temporales o permanentes y capacidad autónoma de desconectar nodos defectuosos.

Todos los nodos pueden ser transmisores y receptores, y la cantidad de los mismos abonados al sistema puede ser variable. Si la situación lo exige, un nodo puede solicitar a otro una determinada información mediante uno de los campos del mensaje (trama remota o RTR).

Cualquier nodo puede introducir una trama en el Bus con la condición de que esté libre, si otro lo intenta al mismo tiempo el conflicto se resuelve por la prioridad del mensaje indicado por el identificador del mismo. El sistema está dotado de una serie de mecanismos que aseguran que el mensaje es transmitido y recibido correctamente.

Cuando un mensaje presenta un error, es anulado y vuelto a transmitir de forma correcta, de la misma forma un nodo con problemas avisa a los demás mediante el propio mensaje. Si la situación es irreversible, dicho nodo queda fuera de servicio pero el sistema sigue funcionando.

2.3.2 Protocolo de comunicaciones.

CAN es un protocolo de comunicaciones serie que soporta control distribuido en tiempo real con un alto nivel de seguridad y multiplexación.

El establecimiento de una red CAN para interconectar dispositivos tiene la finalidad de sustituir o eliminar el cableado. Los sensores, sistemas antideslizantes, etc. se conectan mediante una red CAN a velocidades de transferencia de datos de hasta 1 Mbps.

De acuerdo al modelo de referencia OSI (Open Systems Interconnection), modelo de interconexión de sistemas abiertos, la arquitectura de protocolos CAN incluye cuatro capas: física, de enlace de datos, aplicación y una capa especial para gestión y control del nodo llamada capa de supervisor.

- Capa física: define los aspectos del medio físico para la transmisión de datos entre nodos de una red CAN, los más importantes son niveles de señal, representación, sincronización y tiempos en los que los bits se transfieren al bus. La especificación del protocolo CAN no define una capa física, sin embargo, los estándares ISO 11898 establecen las características que deben cumplir las aplicaciones para la transferencia en alta y baja velocidad.
- Capa de enlace de datos: define las tareas independientes del método de acceso al medio, además debido a que una red CAN brinda soporte para procesamiento en tiempo real a todos los sistemas que la integran, el intercambio de mensajes que demanda dicho procesamiento requiere de un sistema de transmisión a frecuencias altas y retrasos mínimos. En redes multimaestro, la técnica de acceso al medio es muy importante ya que todo nodo activo tiene los derechos para controlar la red y acaparar los recursos. Por lo tanto la capa de enlace de datos define el método de acceso al medio así como los tipos de tramas para el envío de mensajes.

Cuando un nodo necesita enviar información a través de una red CAN, puede ocurrir que varios nodos intenten transmitir simultáneamente. CAN resuelve lo anterior al asignar prioridades mediante el identificador de cada mensaje, donde dicha asignación se realiza durante el diseño del sistema en forma de números binarios y no puede modificarse dinámicamente. El identificador con el menor número binario es el que tiene mayor prioridad.

El método de acceso al medio utilizado es el de Acceso Múltiple por Detección de Portadora, con Detección de Colisiones y Arbitraje por Prioridad de Mensaje (CSMA/CD+AMP, Carrier Sense Multiple Access with Collision Detection and Arbitration Message Priority). De acuerdo con este método, los nodos en la red que

necesitan transmitir información deben esperar a que el bus esté libre (detección de portadora); cuando se cumple esta condición, dichos nodos transmiten un bit de inicio (acceso múltiple). Cada nodo lee el bus bit a bit durante la transmisión de la trama y comparan el valor transmitido con el valor recibido; mientras los valores sean idénticos, el nodo continúa con la transmisión; si se detecta una diferencia en los valores de los bits, se lleva a cabo el mecanismo de arbitraje (espera de tiempo aleatorio).

CAN establece dos formatos de tramas de datos (data frame) que difieren en la longitud del campo del identificador, las tramas estándares (standard frame) con un identificador de 11 bits definidas en la especificación CAN 2.0A, y las tramas extendidas (extended frame) con un identificador de 29 bits definidas en la especificación CAN 2.0B.

Para la transmisión y control de mensajes CAN, se definen cuatro tipos de tramas de datos: remota (remote frame), de error (error frame) y de sobrecarga (overload frame). Las tramas remotas también se establecen en ambos formatos, estándar y extendido, y tanto las tramas de datos como las remotas se separan de tramas precedentes mediante espacios entre tramas (interframe space).

En cuanto a la detección y manejo de errores, un controlador CAN cuenta con la capacidad de detectar y manejar los errores que surjan en una red. Todo error detectado por un nodo se notifica inmediatamente al resto.

- Capa de supervisor: La sustitución del cableado convencional por un sistema de bus serie presenta el problema de que un nodo defectuoso puede bloquear el funcionamiento del sistema completo. Cada nodo activo transmite una bandera de error cuando detecta algún tipo de error y puede ocasionar que un nodo defectuoso pueda acaparar el medio físico. Para eliminar este riesgo el protocolo CAN define un mecanismo autónomo para detectar y desconectar un nodo defectuoso del bus, dicho mecanismo se conoce como aislamiento de fallos.
- Capa de aplicación: Existen diferentes estándares que definen la capa de aplicación; algunos son muy específicos y están relacionados con sus campos de aplicación. Entre las capas de aplicación más utilizadas cabe mencionar CAL (CAN Application Layer) en la que se basa el protocolo CANopen, pero también existen otras como DeviceNet, SDS (Smart Distributed System), OSEK o CANKingdom.

2.3.3 Elementos que componen un sistema CANbus

Todo sistema CANbus que se desee implementar debe estar constituido por los siguientes elementos:

MEDIO FÍSICO

La información circula por dos cables trenzados que unen todas las unidades de control que forman el sistema. Esta información se transmite por diferencia de tensión entre los dos cables, de forma que un valor alto de tensión representa un '1' y un valor bajo de tensión representa un '0'. La combinación adecuada de unos y ceros conforman el mensaje a transmitir.

Como se puede observar en la figura 31, en un cable de CANbus los valores de tensión oscilan entre 0V y 2.25V, por lo que se denomina cable L (Low) y en el otro, el cable H (High) lo hacen entre 2.75V y 5V. En caso de que se interrumpa la línea H o que se derive a masa, el sistema trabajará con la señal de Low con respecto a masa, en el caso de que se interrumpa la línea L, ocurrirá lo contrario. Esta situación permite que el sistema siga trabajando con uno de los cables cortados o comunicados a masa, incluso con ambos comunicados también sería posible el funcionamiento, quedando fuera de servicio solamente cuando ambos cables se cortan. Es importante tener en cuenta que el trenzado entre ambas líneas sirve para anular los campos magnéticos, por lo que no se debe modificar en ningún caso ni el paso ni la longitud de dichos cables.

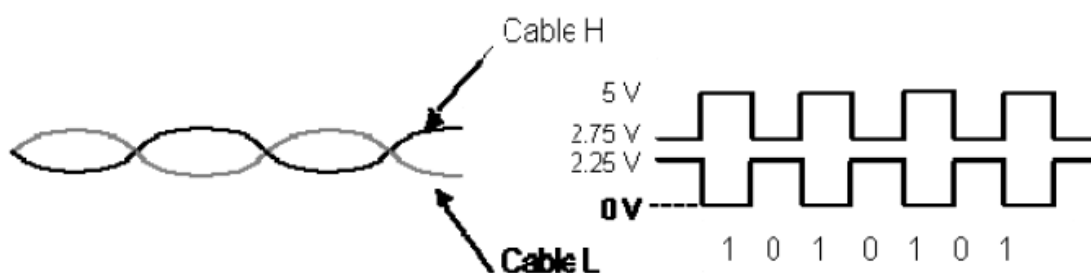


Figura 32. Niveles de tensión de la señal transmitida

ELEMENTO DE CIERRE O TERMINADOR

Son resistencias conectadas a los extremos de los cables H y L. Sus valores se obtienen de forma empírica y permiten adecuar el funcionamiento del sistema a diferentes longitudes de cables y número de unidades de control abonadas, ya que impiden fenómenos de reflexión que pueden perturbar el mensaje. Estas resistencias están alojadas en el interior de algunas de las unidades de control del sistema por cuestiones de economía y seguridad de funcionamiento.

CONTROLADOR

Es el elemento encargado de la comunicación entre el microprocesador de la unidad de control y el transmisor-receptor. Trabaja acondicionando la información que entra y sale entre ambos componentes.

El controlador está situado en el nodo, por lo que existen tantos como unidades estén conectados al sistema. Este elemento trabaja con niveles de tensión muy bajos y es el que determina la velocidad de transmisión de los mensajes, que será más o menos elevada según el compromiso del sistema. Este elemento también interviene en la necesaria sincronización entre las diferentes unidades de mando para la correcta emisión y recepción de los mensajes.

TRANSMISOR / RECEPTOR (TRANSCIVER)

El transmisor-receptor es el elemento que tiene la misión de recibir y de transmitir los datos, además de acondicionar y preparar la información para que pueda ser utilizada por los controladores. Esta preparación consiste en situar los niveles de tensión de forma adecuada, amplificando la señal cuando la información se vuelca en la línea y reduciéndola cuando es recogida de la misma y suministrada al controlador.

El transmisor-receptor es básicamente un circuito integrado que está situado en cada una de las unidades de control abonadas al sistema aunque en ningún caso interviene modificando el contenido del mensaje. Funcionalmente está situado entre los cables que forman la línea CANbus y el controlador. En la figura 32 se muestran todos los componentes que constituyen un nodo de conexión al bus.

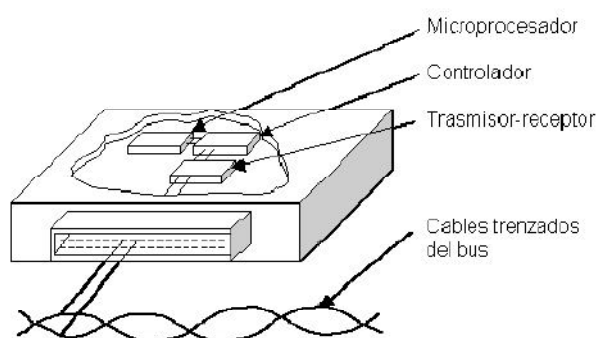


Figura 33. Esquema de sistema conectado a CANbus

2.3.4 Topología del CANbus.

El sistema CANbus está orientado hacia el mensaje. La información en la línea es transmitida en forma de mensajes estructurados en la que una parte del mismo es un identificador que indica la clase de dato que contiene.

Todas las unidades de control reciben el mensaje, lo filtran y solo lo emplean las que necesitan dicho dato. Naturalmente, la totalidad de los nodos abonados al sistema son capaces tanto de introducir como de recoger mensajes de la línea.

Cuando el bus está libre cualquier unidad conectada puede empezar a transmitir un nuevo mensaje. En el caso de que una o varias unidades pretendan introducir un mensaje al mismo tiempo, lo hará la que tenga una mayor prioridad. Esta prioridad viene indicada por el identificador. El proceso de transmisión de datos se desarrolla siguiendo un ciclo de varias fases:

- 1) Suministro de datos: Una unidad de mando recibe información de los sensores que tiene asociados (r.p.m., fuerza, posición, velocidad, distancia, etc.) y su microprocesador pasa la información al controlador donde es gestionada y acondicionada para a su vez ser pasada al transceiver donde se transforma en señales eléctricas.
- 2) Trasmisión de datos: El controlador de dicha unidad transfiere los datos y su identificador junto con la petición de inicio de trasmisión, asumiendo la responsabilidad de que el mensaje sea correctamente transmitido a todos los nodos asociados al sistema. Para transmitir el mensaje ha tenido que encontrar el bus libre, y en caso de colisión intentando transmitir simultáneamente, tener una prioridad mayor. A partir del momento en que esto ocurre, el resto de unidades de mando se convierten en receptoras.
- 3) Recepción del mensaje: Cuando la totalidad de las unidades de mando reciben el mensaje, verifican el identificador para determinar si el mensaje va a ser utilizado por ellos. Las unidades de mando que necesitan los datos del mensaje lo procesan, si no lo necesitan, el mensaje es ignorado.

El sistema CANbus dispone de mecanismos para detectar errores en la transmisión de mensajes, de forma que todos los receptores realizan un chequeo del mensaje analizando una parte del mismo llamado campo CRC. Otros mecanismos de control que se aplican en las unidades emisoras son la monitorización del nivel del bus, la presencia de campos de formato fijo en el mensaje (verificación de la trama), análisis estadísticos por parte de las unidades de mando de sus propios fallos, etc.

Estas medidas hacen que las probabilidades de error en la emisión y recepción de mensajes sean muy bajas, por lo que es un sistema extraordinariamente seguro.

2.3.5 Formato de las tramas CAN.

La información que circula entre las unidades de mando a través del puerto de comunicaciones serie son paquetes de bits con una longitud limitada y con una estructura definida de campos que conforman el mensaje. Uno de los campos actúa de identificador del tipo de dato que se transporta, del nodo que lo trasmite y de la prioridad para transmitirlo respecto a otros. El mensaje no va direccionado ningún nodo en concreto, sino que cada uno de los nodos reconocerá mediante este identificador si el mensaje le interesa o no.

La estructura del mensaje permite llevar a cabo el proceso de comunicación entre las unidades de mando según el protocolo definido por Bosch [8], referencia para el CANbus, donde los distintos campos que lo compone facilitan desde identificar a la unidad de mando, como indicar el principio y el final del mensaje, mostrar los datos, permitir distintos controles, etc.

Los mensajes son introducidos en la línea con una cadencia que oscila entre los 0.055 y 12.9 milisegundos, dependiendo de la velocidad del Bus y de la unidad de mando que los introduce.

La estructura de mensaje, que puede verse en el anexo 9, está compuesto por los siguientes campos:

- Campo de inicio del mensaje (SOF): El mensaje se inicia con un bit dominante '0', cuyo flanco descendente es utilizado por las unidades de mando para sincronizarse entre sí.
- Campo de arbitraje: Los 12 bits de este campo se emplean como identificador y permite reconocer al Nodos que emite la trama, el tipo de mensaje y la prioridad de este. Cuanto más bajo sea el valor del identificador más alta es la prioridad, y por lo tanto determina el orden en el que van a ser introducidos los mensajes en la Bus.
- El bit RTR: Forma parte del campo de arbitraje e indica si el mensaje contiene datos (RTR=0) o si se trata de una trama remota sin datos (RTR=1). Una trama de datos siempre tiene prioridad frente a las tramas remotas que se emplea para solicitar datos a otros nodos, bien porque se necesitan o para realizar un chequeo.
- Campo de control: Este campo informa sobre las características del campo de datos. El bit IDE indica cuando es un '0' que se trata de una trama estándar y cuando es un '1' que es una trama extendida. La diferencia entre una trama estándar y una trama extendida es que el identificador de la primera tiene 11 bits y la segunda 29 bits. Ambas tramas pueden coexistir eventualmente, siempre y cuando todos los controladores del sistema soporten el formato extendido, y la razón de su presencia es la existencia de dos versiones de CAN.

- Campo DLC: Los cuatro bit que lo componen indican el número de bytes contenido en el campo de datos.
- Campo de datos: En este campo aparece la información del mensaje con los datos que la unidad de mando correspondiente introduce en la línea CANbus. Puede contener entre 0 y 8 bytes (de 0 a 64 bit).
- Campo de chequeo (CRC): Este campo tiene un formato predefinido con una longitud de 16 bits de los cuales los 15 primeros son utilizados para la detección de errores, mientras que el último siempre es un bit recesivo '1' que delimita el campo CRC.
- Campo de confirmación (ACK): El campo ACK está compuesto por dos bit que son siempre transmitidos como recesivos '1'. Todos los nodos que reciben el mismo CRC modifican el primer bit del campo ACK por uno dominante '0', de forma que la unidad de mando que está todavía transmitiendo reconoce que al menos uno de los Nodos conectados al sistema ha recibido el mensaje correctamente. De no ser así, el Nodo transmisor interpreta que su mensaje presenta un error.
- Campo de final de mensaje (EOF): Este campo indica el final del mensaje con una cadena de 7 bits recesivos.
- Interframe space (IFS): Está compuesto de al menos tres bits recesivos llamados intermission, y su función es la de separar dos tramas consecutivas. Esto permite el procesamiento interno de los mensajes antes de recibir una nueva trama. Después del IFS el bus permanece en un estado recesivo indicando que está libre para una nueva transmisión.

Puede ocurrir que en determinados mensajes se produzcan largas cadenas de ceros o unos, y que esto provoque una pérdida de sincronización entre los distintos Nodos. El protocolo CAN resuelve esta situación insertando un bit de diferente polaridad cada cinco bits iguales: cada cinco '0' se inserta un '1' y viceversa. El Nodo que utiliza el mensaje, descarta un bit posterior a cinco bits iguales. Estos bits reciben el nombre de bit stuffing.

2.4 Protocolo CANopen

2.4.1 Introducción

El bus de campo CAN solo define la capa física y de enlace del modelo ISO/OSI, por lo que es necesario definir como se asignan y utilizan los identificadores y datos de los mensajes. Para ello se definió el protocolo CANopen [9], que está basado en CAN e implementa la capa de aplicación.

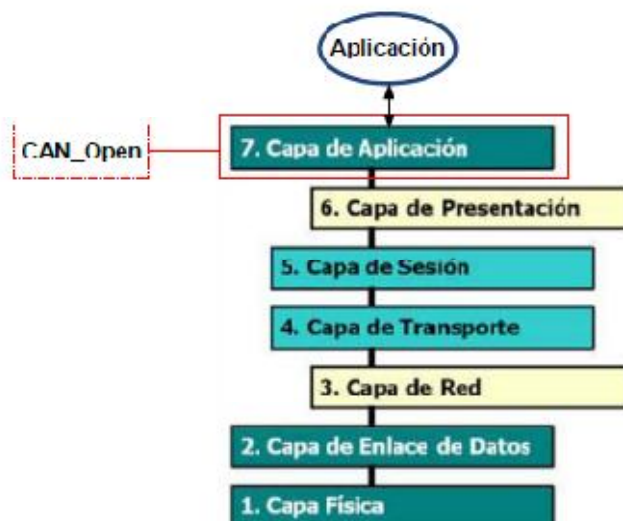


Figura 34. CANopen y modelo de referencia OSI

La capa de aplicación está comprometida con un concepto de configuración, transmisión de datos en tiempo real y el mecanismo de sincronización entre componentes [10]. La funcionalidad que ofrece a la aplicación está dividida en diferentes objetos de comunicación mediante los cuales las aplicaciones interactúan entre sí. Para ello, el objeto de comunicación intercambia datos por el Bus con uno o varios objetos siguiendo un protocolo específico para cada objeto de comunicación.

2.4.2 Modelo del nodo CANopen

La figura 35 muestra la estructura de un nodo CANopen.

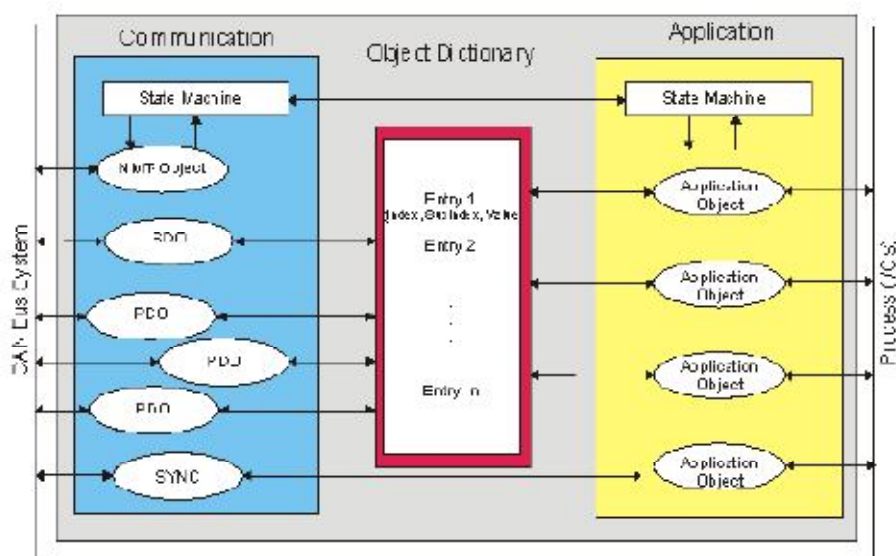


Figura 35. Modelo del nodo.

- **Comunicación:** la función de esta unidad es suministrar los objetos de comunicación y la apropiada funcionalidad para transportar los campos de datos por el Bus.
- **Object Dictionary:** es una colección de todos los campos de datos que influyen en el desarrollo de las aplicaciones de objetos, los objetos de comunicación y la máquina de estados usada en el Nodo.
- **Aplicación:** la aplicación controla la funcionalidad del Nodo con respecto a la interacción con el entorno de procesos.

El diccionario de objetos cumple con la función de interfaz entre la comunicación y la aplicación. La descripción completa de las aplicaciones de un dispositivo con respecto a las entradas en el diccionario del objeto se conoce como perfil del dispositivo.

El diccionario de objetos es una de las partes más importantes del perfil del componente por lo que se tratara en profundidad más adelante.

2.4.3 Diccionario de Objetos

Es un grupo ordenado de objetos. Describe completamente y de forma estandarizada la funcionalidad de cada dispositivo y permite su configuración mediante mensajes a través del propio bus.

Cada objeto se direcciona utilizando un índice de 16 bits. Para permitir el acceso a elementos individuales de las estructuras de datos también existe un subíndice de 8 bits. En la tabla 2 se puede ver la estructura general del diccionario.

Tabla 2. Estructura de un diccionario de objetos estándar en CANopen

INDICE	OBJETO	DESCRIPCION
0x0000	No usado	
0x0001-0x001F	Tipos de dato estáticos	Contiene definiciones de tipos de datos estándar como boolean, enteros, string, punto flotante, etc. Se incluyen como referencia, no pueden ser leídas ni escritas
0x0020-0x003F	Tipos de dato complejos	Contienen definiciones de estructuras predefinidas, compuestas de tipos estático, comunes a todos los dispositivos
0x0040-0x005F	Tipos de dato específicos del fabricante	Contienen definiciones de estructuras predefinidas, compuestas de tipos estático, específicas de un dispositivo en particular
0x0060-0x007F	Tipos de dato estáticos específicos del perfil del dispositivo	Definiciones de tipos de dato básicos específicos para el perfil del dispositivo

0x0080-0x009F	Tipos de dato complejos específicos del perfil del dispositivo	Definiciones de estructuras específicas para el perfil del dispositivo
0x00A0-0x0FFF	Reservado	
0x1000-0x1FFF	Rango para el perfil de comunicaciones	Contiene parámetros de configuración del bus CAN. Estas entradas del diccionario son comunes a todos los dispositivos.
0x2000-0x5FFF	Rango para el perfil específico del fabricante	Contiene las extensiones al perfil estándar, realizadas por el fabricante.
0x6000-0x9FFF	Rango para perfiles de dispositivos estandarizados	Contiene todos los objetos de datos comunes a un tipo de perfil que pueden ser leídos o escritos desde la red. Algunas de las entradas son obligatorias (funcionalidad requerida) mientras que otras son opcionales (funcionalidad opcional)
0xA000-0xFFFF	Reservado	

El rango relevante de objetos va desde el índice 1000 al 9FFF. Para cada nodo de la red existe un OD, diccionario de objetos, que contiene todo los parámetros que describen el dispositivo y su comportamiento en la red.

En CANopen hay documentos que describen perfiles. Hay un perfil de comunicaciones (communication profile) donde están descritos todos los parámetros relacionados con las comunicaciones. Además hay varios perfiles de dispositivos (device profiles) donde se definen los objetos de un dispositivo en particular.

Un perfil define para cada objeto del diccionario su función, nombre, índice, subíndice, tipos de datos, si es obligatorio u opcional, si es de "sólo lectura", "sólo escritura" o "lectura-escritura", etc.

2.4.4 Modelo de comunicaciones

El modelo de comunicaciones de CANopen define cuatro tipos de mensajes (objetos de comunicación):

- **Objetos administrativos:** son mensajes administrativos que permiten la configuración de las distintas capas de la red así como la inicialización, configuración y supervisión de la misma. Se basa en los servicios NMT, LMT y DBT de la capa CAL.
- **Service Data Objects (SDO):** objetos o mensajes de servicio utilizados para leer y escribir cualquiera de las entradas del diccionario de objetos de un dispositivo. Corresponden a mensajes CAN de baja prioridad.

- Process Data Objects (PDO): objetos o mensajes de proceso utilizados para el intercambio de datos de proceso, es decir, datos de tiempo real. Por este motivo, típicamente corresponden a mensajes CAN de alta prioridad.
- Mensajes predefinidos: de sincronización, de emergencia y time stamp. Permiten la sincronización de los dispositivos y generar notificaciones de emergencia en forma opcional.

CANopen soporta los modelos de comunicación punto-a-punto, maestro-esclavo, que podemos observar en la figura 36 y productor-consumidor, en sus variantes push y pull, en la figura 37. En el modelo push los productores colocan los eventos en el canal de eventos y éste se los envía a los consumidores. En el pull el flujo de eventos ocurre en el sentido contrario, es decir, los consumidores solicitan eventos al canal de eventos y éste los solicita a los productores.

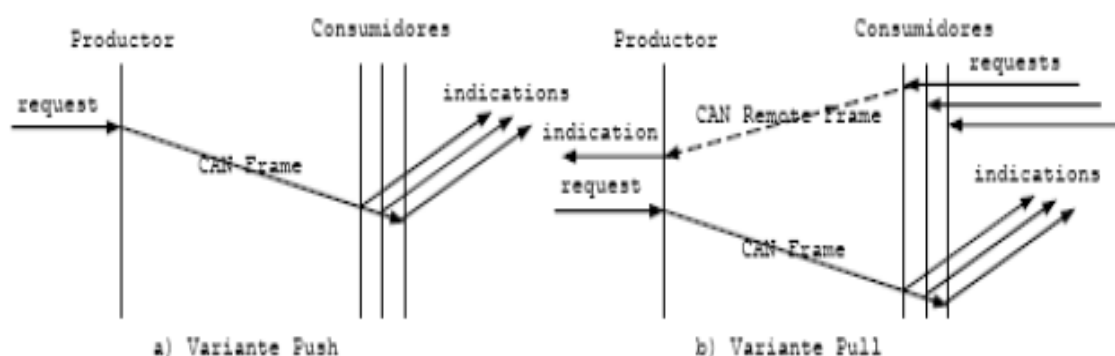


Figura 36. Modelo de comunicación productor-consumidor en CANopen

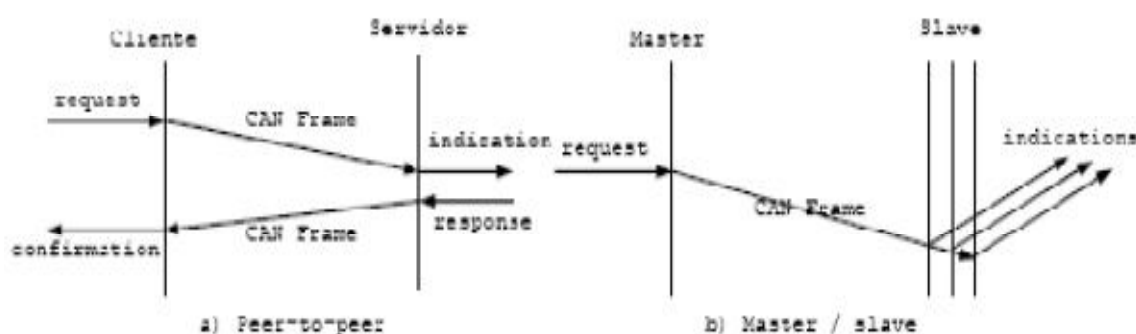


Figura 37. Modelos de comunicación punto-a-punto y maestro-esclavo en CANopen

2.4.5 Identificadores de mensaje

CANopen define la distribución de los identificadores de mensaje (Predefined Connection Set) de manera que hay un mensaje de emergencia por nodo, mensajes de sincronización y time stamp, un SDO, ocupando dos identificadores, mensajes NMT y cuatro PDOs de transmisión y cuatro de recepción por dispositivo.

El identificador de 11 bits, como se puede ver en la figura 38, se divide en dos partes:

- 4 bits para el código de función
- 7 bits para el identificador de nodo (NodeID)

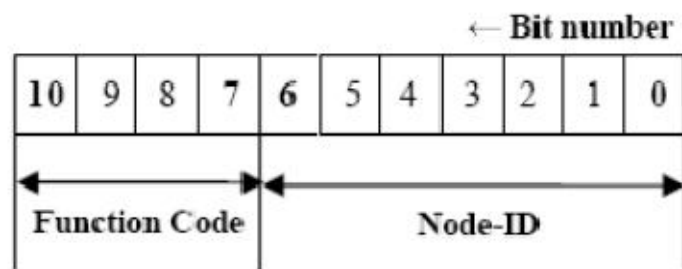


Figura 38. Estructura del identificador de mensajes CAN

La distribución de los identificadores se corresponde con una estructura del tipo maestro-esclavo. El maestro conoce los NodeIDs de todos los esclavos conectados (máximo 127) pero dos esclavos no pueden comunicarse entre sí porque no conocen sus identificadores.

En la tabla 2 se puede ver la distribución general de los identificadores.

Tabla 3. Asignación de los identificadores CAN en CANopen

Broadcast objects of the CANopen Predefined Master/Slave Connection Set			
Object	Function code (ID-bits 10-7)	COB-ID	Communication parameters at OD index
NMT Module Control	0000	000h	—
SYNC	0001	080h	1005h, 1006h, 1007h
TIME STAMP	0010	100h	1012h, 1013h

Peer-to-Peer objects of the CANopen Predefined Master/Slave Connection Set			
Object	Function code (ID-bits 10-7)	COB-ID *	Communication parameters at OD index
EMERGENCY	0001	081h - 0FFh	1024h, 1015h
PDO 1 (transmit)	0011	181h - 1FFh	1800h
PDO 1 (receive)	0100	201h - 27Fh	1400h
PDO 2 (transmit)	0101	281h - 2FFh	1801h
PDO 2 (receive)	0110	301h - 37Fh	1401h
PDO 3 (transmit)	0111	381h - 3FFh	1802h
PDO 3 (receive)	1000	401h - 47Fh	1402h
PDO 4 (transmit)	1001	481h - 4FFh	1803h
PDO 4 (receive)	1010	501h - 57Fh	1403h
SDO (transmit/server)	1011	581h - 5FFh	1200h
SDO (receive/client)	1100	601h - 67Fh	1200h
NMT Error Control	1110	701h - 77Fh	1016h, 1017h

2.4.6 Service Data Objects (SDO)

Normalmente este tipo de objetos es usado para configuración de dispositivos y transferencia de grandes cantidades de datos no relevantes en forma directa para el control del proceso. En comparación con los PDOs, son mensajes de baja prioridad.

Los objetos de servicio o SDOs permiten implementar los modelos de comunicación cliente-servidor o punto-a-punto, para acceder a los diccionarios de objetos de los dispositivos. Para un cierto SDO, el dispositivo cuyo diccionario está siendo accedido es el servidor mientras que el otro dispositivo, el que inicia la actividad, es el cliente. Este tipo de objetos ofrece transferencia de datos sin conexión y con confirmación. Por este motivo, cada SDO involucra el intercambio de dos tramas CAN con diferentes identificadores. Un SDO es representado en CMS como un objeto de tipo Multiplexed Domain.

Se definen una serie de protocolos petición-respuesta que se pueden aplicar a los SDOs para su transferencia:

- Initiate Domain Download
- Initiate Domain Upload
- Download Domain Segment
- Upload Domain Segment
- Abort Domain Transfer Download significa escribir en el diccionario de objetos y upload leer del él.

Al ser Multiplexed Domains desde el punto de vista de CMS, los SDOs pueden transferir datos de cualquier longitud. Sin embargo CANopen define dos tipos de transferencia para los SDOs basándose en el tamaño de los datos a transferir:

- Transferencia expedita: para datos de longitud menor o igual a 4 *bytes*.
- Transferencia en segmentos: para datos de longitud mayor de 4 *bytes*.

Los mensajes tanto del cliente como del servidor siempre tienen una longitud de 8 bytes aunque no todos contengan información significativa. Es importante tener en cuenta que en CANopen los parámetros de más de un *byte* se envían siempre en la forma little endian, es decir, primero el *byte* menos significativo (LSB).

TRANSFERENCIA EXPÉDITA

Usada para transmitir mensajes con longitud de datos menor o igual a 4 *bytes*, para lo cual se usan los protocolos Initiate Domain Download o Initiate Domain Upload.

No se aplica fragmentación, se envía un único mensaje CAN y se recibe la confirmación del servidor.

Un SDO que se transmite del cliente al servidor bajo este protocolo tiene el siguiente formato:

- Command Specifier (1 byte): contiene información sobre si es subida o descarga de datos, petición o respuesta, transferencia expedita o en segmentos, tamaño de los datos y el toggle bit:
 - Client Command Specifier (3 bits): 001 para download y 010 para upload.
 - Ignorado (1 bit): se pone a 0.
 - Número de bytes / Ignorado (2 bits): número de bytes que no contienen datos. Es válido si los siguientes dos bits son 1. Si no, vale 0. En upload se ignora.
 - Transfer Expedited / Ignorado (1 bit): indica si se trata de una transferencia expédita (a 1) o una transferencia en segmentos (a 0). Tiene que estar siempre a 1 para este tipo de transferencia. En upload se ignora.
 - Block Size Indicated / Ignorado (1 bit): si está a 1 es que el tamaño de los datos está especificado, si no, no. En upload se ignora.
- Index (2 bytes): índice de la entrada del diccionario de objetos del servidor que el cliente desea acceder mediante el SDO actual. Este campo y el siguiente forman el multiplexor del dominio.
- Subindex (1 byte): subíndice de la entrada del diccionario de objetos del servidor que el cliente desea acceder. Sólo tiene sentido si la entrada es de un tipo complejo. Si se trata de una entrada con tipo estático, debe ser 0.
- Datos / Ignorado (4 bytes): datos que se desean enviar al servidor (el valor de una entrada en el diccionario si se está escribiendo). Si es upload se ignora.

El servidor, al recibir el mensaje anterior, y si ha accedido con éxito al diccionario, contesta con un mensaje con el siguiente formato:

- Command Specifier (1 byte):
 - Server Command Specifier (3 bits): 011 para download y 010 para upload.
 - Ignorado (1 bit): se pone a 0.
 - Ignorado / Número de bytes (2 bits): si es un download se ignora.
 - Ignorado / Transfer Expedited (1 bit): si es un download se ignora.
 - Ignorado / Block Size Indicated (1 bit): si es un download se ignora.
- Ignorado / Datos (4 bytes): si es un download (escritura) se ignora ya que es una confirmación. Si es una lectura o upload, contiene el valor leído del diccionario de objetos del servidor.

TRANSFERENCIA EN SEGMENTOS

Usada para transmitir mensajes con longitud de datos mayor de 4 bytes. Se aplica fragmentación en segmentos partiendo los datos en múltiples mensajes CAN. El cliente espera confirmación del servidor por cada segmento.

Para el primer mensaje tanto del cliente como del servidor, se usan los protocolos Initiate Domain Download o Initiate Domain Upload según corresponda, con el bit Transfer Expedited a '0' (para transferencia en segmentos). Si ese bit está a '0' y el siguiente (Block Size Indicated) está a '1', significa que el campo de datos contiene el número de bytes que se van a transmitir (al ser little endian el byte 4 del mensaje contiene el LSB y el 7 el MSB).

Para los mensajes siguientes se usan los protocolos Download Domain Segment o Upload Domain Segment, según se quiera leer o escribir una entrada en el diccionario. Los mensajes CAN bajo estos dos protocolos tienen el siguiente formato:

- Command Specifier (1 byte):
 - Client Command Specifier (3 bits): 000 para download y 011 para upload.
 - Toggle Bit (1 bit): es un bit que vale 0 o 1 de forma alternada en segmentos consecutivos. La primera vez vale 0. Teniendo en cuenta que el mecanismo de intercambio sólo permite un mensaje pendiente de confirmación, un único bit es suficiente para esto. El servidor se limita a hacer un eco del valor recibido.
 - Número de bytes / Ignorado (3 bits): indica el número de bytes que no contienen datos, o cero si no especifica el tamaño. Se ignora en el upload.
 - Last Segment Indication / Ignorado (1 bit): vale 1 si se trata del último segmento del SDO que se está transmitiendo y 0 si hay más segmentos. Para upload se ignora.
- Datos / Ignorado (7 bytes): datos que se desean enviar al servidor. Son los bytes que no caben en el mensaje CAN inicial. Si es upload se ignora.

El servidor, al recibir el mensaje anterior, contesta con un mensaje con el formato:

- Command Specifier (1 byte):
 - Client Command Specifier (3 bits): 001 para download y 000 para upload.
 - Toggle Bit (1 bit): igual que antes. Ignorado / Número de bytes (3 bits): se ignora en el download. Ignorado / Last Segment Indication (1 bit): se ignora en el download.

- Ignorado / Datos (7 bytes): si es un download (escritura) se ignora porque es una confirmación. Si es una lectura o upload, contiene el valor leído del diccionario de objetos del servidor.

ABORT DOMAIN TRANSFER

Existe la posibilidad que al acceder a las entradas del diccionario de objetos del servidor se produzca un error. Este protocolo es usado en esos casos para notificar tanto a clientes como a servidores.

El formato de los mensajes de este protocolo es el siguiente:

- Command Specifier (1 byte):
 - Command Specifier (3 bits): 100 para Abort Domain Transfer.
 - Ignorado (5 bits): se ignoran.
- Index (2 bytes): índice de la entrada del diccionario de objetos del servidor que causó el error que se está notificando. Este campo y el siguiente forman el multiplexor del dominio.
- Subindex (1 byte): subíndice de la entrada del diccionario de objetos del servidor que causó el error que se está notificando. Sólo tiene sentido si la entrada es de un tipo complejo. Si se trata de una entrada con tipo estático, debe ser '0'.
- Código de error (4 bytes): código que identifica el error. En la tabla 4 se pueden ver sus posibles valores.

Tabla 4. Códigos de error para SDO Abort Domain Transfer

Abort code	Description
0503 0000	Toggle bit no alternated
0504 0000	SDO protocol timed out
0504 0001	Client/Server command specifier not valid or unknown
0504 0002	Invalid clock size (Block Transfer mode only)
0504 0003	Invalid sequence number (Block Transfer mode only)
0503 0004	CRC error (Block Transfer mode only)
0503 0005	Out of memory
0601 0000	Unsupported Access to an object
0601 0001	Attempt to read a read-only object
0601 0002	Attempt to write a read-only object
0602 0000	Objects does not exist in the Object Dictionary
0604 0041	Object can not be mapped to the PDO
0604 0042	The number and lengh of the objects to be mapped would exceed PDO lengh
0604 0043	General parameter incompatibility reason
0604 0047	General internal incompatibility in the device
0606 0000	Object Access failed due to a hardware error
0606 0010	Data type does not match, lengh of service parameter does not match

0606 0012	Data type does not match, lengh of service parameter is too big
0606 0013	Data type does not match, lengh of service parameter is too low
0609 0011	Sub-index does not exist
0609 0030	Value range of paramenter exceeded (only for write access)
0609 0031	Value of parameter is too high
0609 0032	Value of parameter is too low
0609 0036	Maximum value is less tan mínimum value
0800 0000	General error
0800 0020	Data can not be tranferred or stored to the application
0800 0021	Data can not be tranferred or stored to the application because of local control
0800 0022	Data can not be tranferred or stored to the application because of the present device state
0800 0023	Object Dictionaty dynamic generation fails or no Object Dictionary is present (eg. OD is generate from file and generation fails because of a file error)

Los SDOs requieren ser definidos en el diccionario de objetos mediante una estructura que contiene parámetros relacionados con la transmisión de los mismos. La estructura para el primer SDO para servidores tiene un índice de 0x1200 mientras que el primer SDO para clientes, se ubica en la entrada 0x1280. En total, en una red CANopen, se pueden definir hasta 128 SDOs para clientes y 128 SDOs para servidores.

2.4.7 Process Data Objects (PDO)

Este tipo de objetos permite intercambiar datos del proceso en tiempo real. Implementa el modelo de comunicaciones productor-consumidor. Los datos se transmiten desde un productor a varios consumidores. Ofrece un servicio de transferencia de datos sin conexión y sin confirmación. No se aplica un protocolo de fragmentación y reensamble de los objetos. Los PDOs están pensados para tráfico de tiempo real de alta prioridad, por lo que es conveniente evitar la sobrecarga que produciría agregar un protocolo de fragmentación y confirmación como el que se usa en los SDOs.

Los TPDO son aquellos mensajes con información del proceso que el nodo transmite (por ejemplo la lectura de un sensor). Por otro lado, los RPDO son los mensajes con información del proceso que el nodo escucha (por ejemplo un nodo que controle la apertura de una válvula escuchará el bus en busca de órdenes). El contenido de un PDO está definido tan sólo por su identificador. Tanto el emisor como el receptor deben conocerlo para poder interpretar su estructura interna.

Cada PDO se describe mediante dos objetos del diccionario:

- PDO Communication Parameter: contiene el COB-ID que utiliza el PDO, el tipo de transmisión, tiempo de inhibición y temporizador.

- PDO Mapping Parameter: contiene una lista de objetos del OD contenidos en la PDO, incluyendo su tamaño en bits.

CANopen define varios mecanismos de comunicación para la transmisión de PDOs:

- Transmisión asíncrona:
 - Eventos: la transmisión de un mensaje es causada por la ocurrencia de un evento específico definido en el perfil del dispositivo.
 - Temporizador: existe un temporizador que cada cierto tiempo cause la transmisión.
 - Solicitud remota: la transmisión asíncrona de mensajes PDO puede comenzar al recibir una solicitud remota (trama RTR) enviada por otro dispositivo.
- Transmisión sincrónica: la transmisión sincrónica de mensajes PDO es disparada por la expiración de un período de transmisión, sincronizado mediante la recepción de objetos SYNC. Es decir, cada vez que llega un mensaje SYNC, se abre una ventana de transmisión sincrónica. Los PDOs sincrónicos deben ser enviados dentro de esa ventana. Se distinguen dos modos dentro de este tipo de transmisión:
 - Modo cíclico: son mensajes que se transmiten dentro de la ventana abierta por el objeto SYNC. No se transmiten en todas las ventanas sino con cierta periodicidad, especificada por el campo Transmission Type del Communication Parameter correspondiente.
 - Modo acíclico: son mensajes que se transmiten a partir de un evento de la aplicación. Se transmiten dentro de la ventana pero no de forma periódica.

En la tabla 5 se muestran los distintos modos de transmisión de PDOs, definidos por el Transmission Type (entero de 8 bits) del Communication Parameter.

Tabla 5. Modos de transmisión de PDOs en CANopen

Transmission Type	Condition to trigger PDO (B-both needed, O-one or both)			PDO Transmission
	SYNC	RTR	EVENT	
0	D	-	B	Sync acyclic
1-240	O	-	-	Sync cyclic
241-251	-	-	-	reserved
252	B	B	-	Sync after RTR
253	-	O	-	Async after RTR
254	-	O	O	Async manufacture specific event
255	-	O	O	Async, device profile specific event

*SYNC = objeto SYNC recibido

*RTR = recibida trama RTR

*Event = cambio de valor de un dato, temporizador...

Un PDO puede tener asignado un tiempo de inhibición que define el tiempo mínimo que debe pasar entre dos transmisiones consecutivas del mismo PDO. Forma parte del Communication Parameter. Está definido como un entero de 16 bits en unidades de 100 microsegundos.

2.4.8 Mensajes adicionales

MENSAJE DE TIME STAMP

Este tipo de objetos representan una cantidad absoluta de tiempo en milisegundos. Proporciona a los dispositivos un tiempo de referencia común. La etiqueta temporal o time-stamp se implementa como una secuencia de 48 bits.

MENSAJE DE SINCRONIZACIÓN (SYNC)

En una red CANopen, hay un dispositivo que es el productor de objetos SYNC y una serie de dispositivos consumidores de objetos SYNC. Cuando los consumidores reciben el mensaje del productor, abren su ventana de sincronismo y pueden ejecutar sus tareas sincrónicas. Este mecanismo permite coherencia temporal y coordinación entre los dispositivos. Por ejemplo, un conjunto de sensores pueden leer las variables del proceso controlado en forma coordinada y obtener así una imagen consistente del mismo.

El COB-ID usado por este objeto de comunicación puede encontrarse en la entrada 0x1005 del diccionario. Para garantizar el acceso de estos objetos al bus, debería asignárseles un COBID bajo. El conjunto predefinido de conexiones de CANopen sugiere usar un valor de 128. El campo de datos del mensaje CAN de este objeto se envía vacío.

Como se puede observar en la figura 39 el comportamiento de estos mensajes viene determinado por dos parámetros (longitud de ventana y período de transmisión).

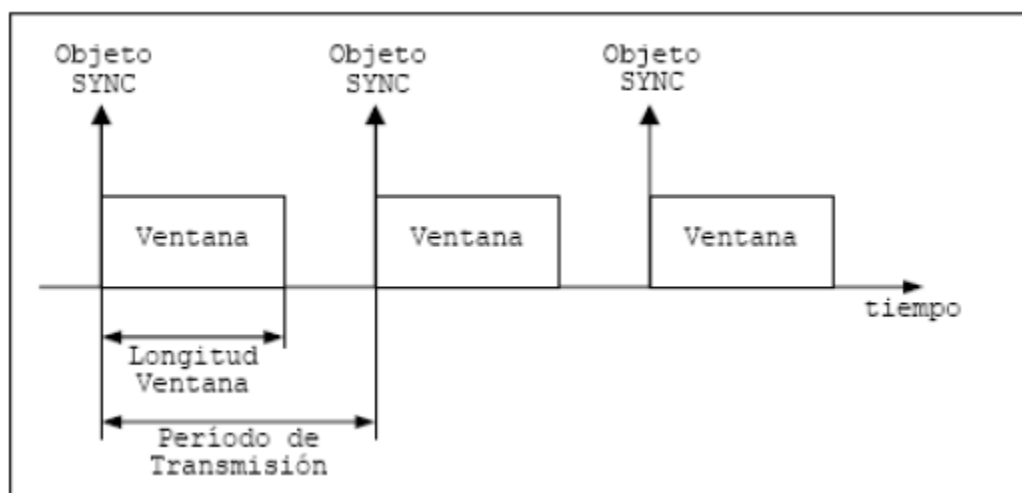


Figura 39. Parámetros de un objeto SYNC en CANopen

La longitud de la ventana o Synchronous Window Length puede ubicarse en la entrada 0x1007 del diccionario. El período de transmisión se encuentra en la posición 0x1006.

MENSAJE DE EMERGENCIA

Estos mensajes se envían cuando ocurre un error interno en un dispositivo. Se transmiten al resto de dispositivos con la mayor prioridad. Pueden usarse como interrupciones o notificaciones de alertas.

Un mensaje de emergencia tiene 8 bytes. En la tabla 6 se muestra su estructura.

Tabla 6. Estructura de un mensaje de emergencia en CANopen

COB-ID	Byte 0-1	Byte 2	Byte 3
0x080 + NODE_ID	Emergency error code	Error register (Object 0x1001)	Manufacturer specific error field

- **Emergency Error Code (2 bytes):** código del error que causó la generación del mensaje de emergencia. Se trata de fallos internos de los dispositivos por lo cual, los errores se relacionan con fallos de tensión, de corriente, del software del dispositivo, del adaptador del bus CAN, etc. En la tabla 7 se muestran los códigos correspondientes en hexadecimal.

Tabla 7. Códigos de error para los mensajes de emergencia de CANopen

Emergency error code	Meaning
00xx	Error reset or NO error
10xx	Generic error
20xx	Current
21xx	Current, device input side
22xx	Current inside the device
23xx	Current device output side
30xx	Voltage
31xx	Mains voltage
32xx	Voltage inside the device
33xx	Out put voltage
40xx	Temperature
41xx	Ambient temperatura
42xx	Device temperature
50xx	Device hardware
60xx	Devoce software
61xx	Internal software
62xx	User software
63xx	Data set
70xx	Additional modules
80xx	Monitoring
81xx	Communication
8110	CAN overrun
8120	Error passive
8130	Life guard error or heart beat error
8140	Recovered from Bus off
82xx	Protocol erro
8210	PDO no processed due to length error
8220	Length exceeded
90xx	External error
F0XX	Additional funtions
FFXX	Device specific

'xx' es la parte dependiente del perfil del dispositivo

- Error Register (1 byte): entrada con índice 0x1001 del diccionario de objetos. Cada bit de este registro indica una condición de error distinta cuando está a '1'. En la tabla 8 se observa el significado de cada bit.

Tabla 8. Bits del *Error Register* de los mensajes de emergencia de CANopen

Bit	Significado
0	Error genérico
1	Problema de corriente
2	Problema de tensión
3	Problema de temperatura
4	Error de comunicación
5	Específico del perfil de dispositivo
6	Reservado
7	Específico del fabricante del dispositivo

- Manufacturer-specific Error Field (5 bytes): este campo puede usarse para información adicional sobre el error. Los datos incluidos y su formato son definidos por el fabricante del dispositivo.

MENSAJES DE NODE/LIFE GUARDING

Se utiliza para saber si un nodo está operativo. La comunicación se basa en el concepto de maestro-esclavo. El NMT maestro monitorea el estado de los nodos. A esto se le llama node guarding. Opcionalmente los nodos pueden monitorear el estado del NMT maestro. A esto se le llama life guarding. Comienza en el NMT esclavo después de haber recibido el primer mensaje de node guarding del NMT maestro. Sirve para detectar errores en las interfaces de red de los dispositivos, pero no fallos en los dispositivos en sí (ya que estos son avisados mediante mensajes de emergencia).

Se puede implementar de dos maneras distintas, pudiendo utilizarse sólo una de ellas a la vez: NMT Node Guarding o Heartbeat.

NMT NODE GUARDING

El maestro va preguntando a los esclavos si están “vivos” cada cierto tiempo. Si alguno no contesta en un tiempo determinado significa que está caído y se informa de ello. Si los esclavos también monitorean al maestro deben informar de que el maestro está caído si no reciben mensajes de Node Guarding de él durante un determinado intervalo. El maestro interroga a los esclavos mediante una trama remota (RTR) con la estructura mostrada en la tabla 9.

Tabla 9. Trama RTR que el NMT maestro envía a los NMT esclavos

COB-ID
0x700 + Node_ID

En la tabla 10 se puede ver el mensaje con el que los NMT esclavos responden a éste.

Tabla 10. Mensaje que los NMT esclavos envían al NMT maestro

COB-ID	Byte 0
0x700 + Node_ID	bit 7: <i>toggle</i> , bit 6-0: <i>state</i>

El toggle bit (bit 7) es un bit que va alternando su valor en cada mensaje de Node Guarding. La primera vez vale '0'. Los bits del 0 al 6 indican el estado del nodo. En la tabla 11 se describen sus valores correspondientes.

Tabla 11. Valor del campo state en un NMT Node Guarding Heartbeat

Value	State
0	Initialising
1	Disconnected *
2	Connecting *
3	Preparing *
4	Stopped
5	Operational
127	Pre-operational

Los estados marcados con * son por si se realiza un *boot-up* extendido

Cada nodo manda un mensaje de Hearbeat cada cierto tiempo para informar de que está operativo. En este caso el mensaje de Boot-up se considera que es el primer mensaje de Heartbeat. Si el NMT maestro deja de recibir estos mensajes durante un tiempo determinado significará que el nodo está caído.

En la tabla 12 se muestra la estructura de este tipo de mensaje, y en la tabla 13 se describe el estado (state) en que se encuentra el nodo.

Tabla 12. Estructura de un mensaje de Heartbeat

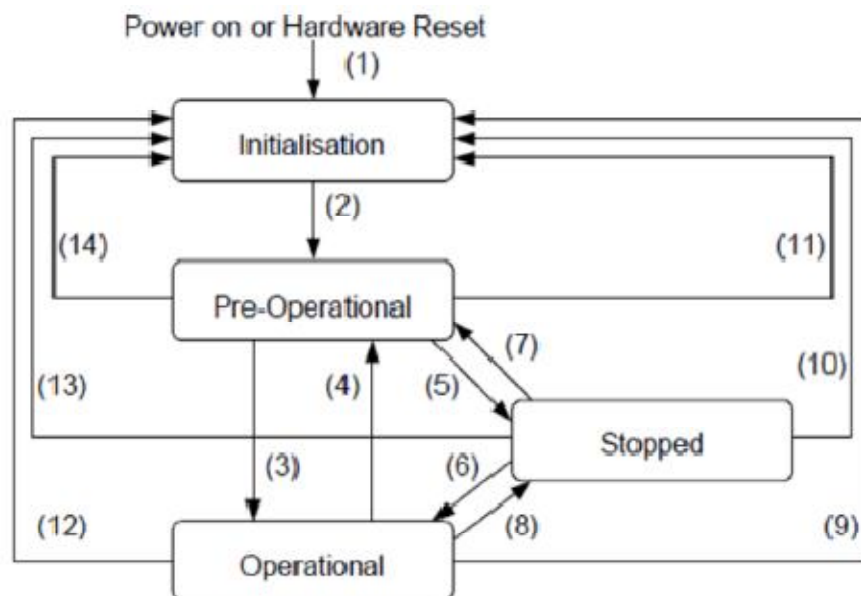
COB-ID	Byte 0
0x700 + Node_ID	<i>state</i>

Tabla 13 Valor del campo state en un mensaje de Heartbeat

state	Meaning
0	Boot-up
4	Stopped
5	Operational
127	Pre-operational

2.4.9 Máquina de Estados.

Todo dispositivo CANopen implementa una máquina de estados que describe los estados posibles en los que se puede encontrar y las secuencias de control posibles para pasar de un estado a otro, tal y como se muestra en la figura 40.



(1)	Tras la activar el dispositivo entra en el estado INITIALISATION automáticamente.
(2)	Inicialización finalizada - entra en PRE-OPERATIONAL automáticamente.
(3),(6)	Servicio Start_Remote_Node.
(4),(7)	Servicio Enter_PRE-OPERATIONAL.
(5),(8)	Servicio Stop_Remote_Node.
(9),(10),(11)	Servicio Reset_Node.
(12),(13),(14)	Servicio Reset_Communication.

Figura 40: Máquina de estados de un dispositivo CANopen.

A continuación se describen los cuatro estados:

INITIALISATION:

Este estado es donde el nodo entra después de activar el dispositivo ó resetearlo por hardware y es donde los valores de áreas del perfil de fabricación, los valores de área de dispositivo estandarizado y los parámetros del área de comunicaciones recuperan sus valores predefinidos. Tras la inicialización se envía el mensaje de Bootup para pasar al estado siguiente.

PRE-OPERATIONAL:

En este estado la configuración de PDOs, los parámetros del dispositivo y la asignación de objetos de aplicación (PDO-maping) pueden ser realizados por medio de SDOs. El dispositivo puede pasar directamente al estado OPERATIONAL por medio del servicio Start Remote Node.

OPERATIONAL:

En este estado todos los objetos de comunicación están activados y es el estado normal de funcionamiento de un nodo CANopen.

STOPPED:

Cuando el dispositivo entra en este estado se detiene toda la comunicación a excepción del node guarding ó heartbeat, si están activos, y los servicios de control de módulos del NMT.

Capítulo 3

Diseño

En este capítulo se explicara con detalle los pasos seguidos para el diseño e implementación del sistema anticolidión con sensores de proximidad sin contacto. Se tratarán en detalle todas las especificaciones del sensor que se va a utilizar, la selección de componentes y los pasos seguidos para crear el diseño del sistema encargado del tratamiento y transmisión de datos.

3.1 Requisitos del sistema

El sistema anticolidión a desarrollar es un sistema que permitirá al ASIBOT evitar colisiones indeseadas con objetos cercanos a la trayectoria en la que se desplaza o que se encuentren cerca de otros objetos que el brazo intenta alcanzar.

El ASIBOT es un brazo robot de cinco grados de libertad que servirá de ayuda a personas, con algún tipo de discapacidad, en la manipulación de objetos y herramientas.

Los seres humanos poseemos el sentido de la vista para orientarnos y este nos permite evitar objetos que se encuentran en nuestro camino y alcanzar los que deseamos. Esto es a grandes rasgos lo que se pretende conseguir con el sistema anticolidión del que dotaremos al ASIBOT. A grandes rasgos porque nuestro sistema no identificara formas, colores, texturas, etc... sino que solo detectara la presencia de objeto o no a cierta distancia.

El sistema anticolidión debe adaptase al diseño del ASIBOT y debe cumplir una serie de requisitos que vienen determinados por la configuración electrónica y por cómo se quiere detectar el objeto y cómo se desee actuar ante éste.

En base a la estructura que tiene el ASIBOT se ha ideado que el sistema anticolidión tenga forma circular y que vaya localizado en la pinza, a modo de pulsera, de modo que durante el desplazamiento hacia un objeto concreto esta pulsera permita la detección de otros objetos cercanos.

Debido al emplazamiento de la pulsera anticolidión, ésta debe ser ligera para que no sea un lastre en el movimiento del brazo.

La idea es que el sistema anticolidión sea portátil en el robot, para así poder colocarse en el lugar que consideremos adecuado en cada momento, o incluso poder colocar varios.

Debido al diseño del ASIBOT el sistema anticolidión completo, definiéndose completo como el anillo de sensores mas circuito necesario para la interpretación de las señales que estos proporcionen, debe estar distribuido a lo largo del brazo con el fin de ocupar el menor espacio posible y evitando de esta manera la concentración de un bloque en un punto del brazo.

No solo existen requisitos físicos en el desarrollo del sistema anticolidión, que están definidos por la estructura del robot y por el diseño ideado, sino que el robot lleva tiempo funcionando en el laboratorio y a la hora de implementar un aplicación nueva debemos adaptarnos a la configuración electrónica con la que trabaja.

El ASIBOT sabemos que está alimentado con 24 V, luego debemos tener en cuenta a la hora de elegir sensores y resto de componentes que ninguno supere este umbral de alimentación y adaptar la alimentación para los componentes que trabajen a menor voltaje.

3.2 Selección de componentes.

La elección del sensor es una de las partes más importantes del diseño. Un profundo análisis realizado sobre los diferentes tipos de sensores de proximidad sin contacto y estudio de diferentes proyectos que incorporan un sistema anticolidión ha sido determinante para la elección del tipo de sensor. El sensor seleccionado va a condicionar todos los parámetros relacionados con la adquisición de datos, como pueden ser la lectura de la distancia, precisión, velocidad de lectura, fiabilidad, etc. Otra de las características a tener muy en cuenta es el modo de lectura de los datos que proporcione dicho sensor y la posibilidad de conexión con los demás elementos que formen el Nodo CANopen.

Teniendo en cuenta lo mencionado anteriormente se ha seleccionado el sensor infrarrojo HDSL-9100 de Agilent Technologies, que podemos observar en la figura 41. Este sensor infrarrojo además de caracterizarse por su reducido tamaño, su bajo consumo y coste, permite una fácil adaptación a nuestro diseño. El HDSL-9100 proporciona una lectura de la distancia mediante una señal analógica en forma de voltaje y además permite la detección de objetos en un rango de distancias, en el que es imposible trabajar con otros sensores infrarrojos, un rango que oscila entre los 5mm y 60mm que pueden ampliarse hasta 150/200mm mediante un acondicionador de señal.



Figura 41. Sensor infrarrojo HDSL-9100

Una vez que tenemos claro cuál es el sensor que se va a utilizar, el siguiente paso es seleccionar todos los componentes que posibilitaran la lectura del mismo y la integración en un Bus CANopen. Estos elementos son los siguientes:

- Acondicionador de señal APDS 9700
- Multiplexor 2 a 1
- Microcontrolador
- Controlador CANbus
- Transmisor-Receptor(Transceiver)

En cuanto a la lectura de sensor, el HDSL-9100, proporciona una señal analógica que puede tratarse por los puertos ADC del microprocesador, por lo que no va a ser un factor determinante en la elección del micro. Lo mismo pasaría con el controlador CAN y el transceiver, ya que son componentes estandarizados y que muchas empresas suministran con unos costes muy similares. Por lo que el factor determinante a la hora de decidir que modelos serian utilizados, fue el soporte que los fabricantes de estos componentes ponen a disposición de los diseñadores de estos sistemas y en este sentido Microchip® es una gran opción.

Microchip® además de ofrecer una completa línea de productos para el diseño de aplicaciones embebidas usando el protocolo CAN, como lo es la Application Note AN945 [11], también ofrece un entorno de desarrollo y programación muy potente como es MPLAB IDE y la herramienta MPLAB C Compiler C18 [12], que incorpora una amplia gama de librerías y funciones predefinidas que facilitan y simplifican de manera significativa el desarrollo de la aplicación deseada.

A la hora de seleccionar los controladores, o también llamados *transceivers*, que gestionan el CAN-Bus y la comunicación serie no se encuentran grandes problemas ya que éstos son componentes muy comunes en el mercado, con diferentes fabricantes

que ofrecen prestaciones similares. Dentro de las opciones que Microchip pone a disposición del diseñador a través de su línea de productos CAN, para este diseño concreto se ha optado seleccionar un microcontrolador que integre un módulo CAN y unos controladores de comunicaciones. Con esto se simplifica la parte hardware ya que no es necesario integrar en la placa del diseño un controlador CAN, ya que este viene incorporado en el microcontrolador. Se simplifica la parte software ya que no es necesario programar el puerto serie que comunica el microcontrolador con el controlador CAN. Se reducen posibles problemas derivados de la inclusión de otro componente en la placa (conexiones, ruidos, etc.), y se reduce el coste total de los componentes.

Teniendo en cuenta que entre las características del microcontrolador han de estar presentes al menos 8 puertos ADC para poder tratar la señal analógica de los 16 sensores, que formaran el anillo anticolisión, previo multiplexado, e incorporar un módulo CAN se ha seleccionado el modelo PIC18F2580. Que además incorpora una memoria de programa de 32K, un TIMER de 8 bits y 3 de 16 bits. Características más que suficientes para afrontar el diseño del Nodo CAN.

3.2.1 El Sensor

El sensor seleccionado es el HDSL-9100, un sensor infrarrojo de reflexión directa, constituido por un emisor, que consiste en un led infrarrojo y un fotodiodo que actúa como receptor. El emisor emite pulsos de luz infrarroja que viaja a través del medio, en este caso el aire. Si el fotodiodo no recibe el reflejo de la luz infrarroja implica que no se ha detectado ningún objeto. Por el contrario el receptor recibirá el reflejo de la luz infrarroja en caso de que un objeto se interponga en la dirección de desplazamiento de la misma.

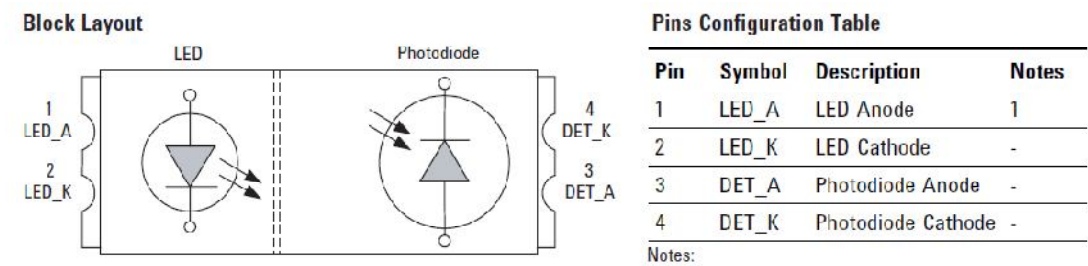


Figura 42. Esquema del sensor y pinout

En la figura 42 se puede ver un esquema del sensor. El pinout del sensor es muy sencillo y como se puede observar se constituye de los ánodos y cátodos del emisor y receptor.

El pin 1 (LED_A) es el ánodo del led infrarrojo (emisor) y estará conectado a una señal PWM producida por un circuito externo que polariza el led para que emita pulsos de luz infrarroja que permitan la detección del objeto.

El pin 2 (LED_K) es el cátodo del emisor y se conecta a GND para hacer posible la polarización del emisor mediante la señal PWM.

El pin 3 (DET_K) es el cátodo del receptor y debido a que un fotodiodo se polariza en inversa es el pin que respecto de tierra nos proporcionara la señal analógica de salida del sensor para que sea tratada y procesada posteriormente.

El pin 4 (DET_A) es el ánodo del fotodiodo y, como se ha dado a entender anteriormente, está conectado a GND para permitir la polarización inversa del mismo.

En la siguiente tabla 14 se presentan las principales características eléctricas del HDSL-9100 seleccionado, destacando especialmente su bajo consumo y intensidad de polarización del led infrarrojo emisor.

Tabla 14. Características eléctricas del HDSL-9100

Parameter	Symbol	Min.	Max	Units		
Emitter						
Continuous Forward Current	I _{DC}	-	50	mA		
Coupled						
Total Power Dissipation (refer to Figure 1)	P _{TOT}	-	165	mW		
Electrical-Optical Characteristics (Ta=25°C)						
Parameter	Symbol	Test Condition	Ratings			Units
			Min	Typ	Max	
Emitter						
Forward Voltage	V _F	I _F = 100mA	—	1.50	1.65	V
Reverse Voltage	V _R	I _R = 10μA	5	—	—	V
Peak Wavelength	λ _p	I _F = 20mA	—	940	—	nm
Spectrum Width of Half Value	Δ _p	I _F = 20mA	—	50	—	nm

Las características principales por las que se ha seleccionado este sensor son:

- Excelente aislamiento óptico
- Alta eficiencia del emisor y alta sensibilidad del fotodiodo receptor
- Bajo coste
- Tamaño reducido
 - Height – 2.70 mm
 - Width – 2.75 mm
 - Length – 7.10 mm
- Se puede combinar con un acondicionador de señal

- Permite la detección de objetos en un desde un rango cercano a 5mm hasta aproximadamente 60mm sin circuito acondicionador

El sensor proporciona una señal analógica en forma de voltaje en función de la distancia a la que se encuentra el objeto. En la figura 43 podemos observar la curva Voltaje vs Distancia del HDSL-9100 dependiendo de la intensidad del pulso con la que se polarice el emisor.

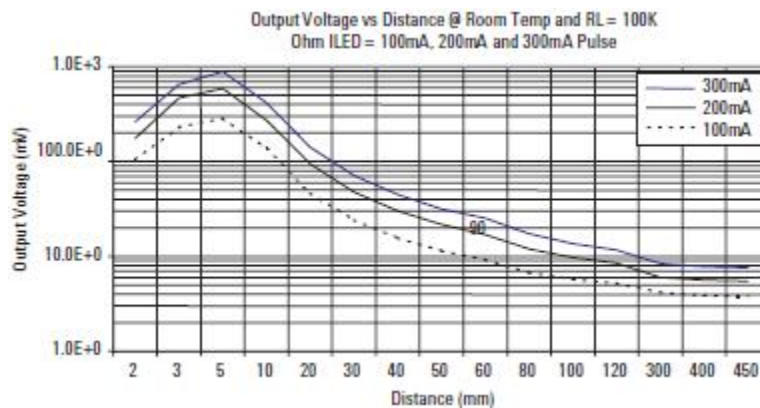


Figura 43. Grafica Voltaje vs Distancia

3.2.2 Circuito acondicionador de señal

El acondicionador de señal seleccionado ha sido el APDS-9700. Es un acondicionador de señal que aumenta la robustez y rendimiento de los sensores ópticos utilizados para la detección de la proximidad o de objetos.

Las características principales de este acondicionador de señal son:

- Bajo consumo de energía
- Control de ancho de pulso
- Control de transmisión de corriente
- Modo de apagado completo
- Tensión de alimentación entre 2,4V a 3,6V
- Permite operar con sensores ópticos en presencia de luz solar
- Inmunidad a la luz artificial
- Dispone de salidas digitales y analógicas
- Flexibilidad para mejorar la distancia de detección hasta 200mm
- Tamaño reducido, 2mm x 2mm, que permite su perfecta introducción en el circuito sensor

Como se puede observar en la figura 44 el acondicionador de señal se compone fundamentalmente de un amplificador de trans-impedancia (TIA), un circuito de

cancelación luz del sol, un convertidor tensión / corriente (V-I) y un comparador de histéresis.

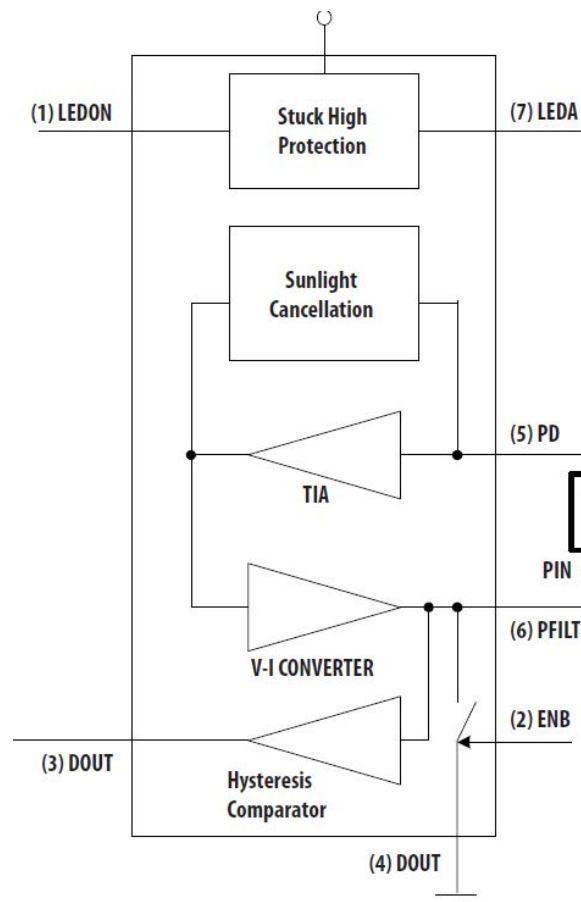


Figura 44. Esquema eléctrico acondicionador de señal APDS-9700

El TIA es regulado internamente y se encarga de convertir la fotocorriente detectada en tensión y amplificar ese voltaje una ganancia predefinida. Se requiere un rechazo a la luz ambiental para el correcto funcionamiento del sensor óptico. El circuito de cancelación de luz solar se utiliza para eliminar el efecto de la luz solar intensa en forma de corriente continua cuando el dispositivo se expone a la luz solar. Este circuito de cancelación también elimina el parpadeo de 50/60 Hz emitido por una fuente de luz artificial. Por tanto el circuito de cancelación proporciona gran fiabilidad al sensor óptico cuando este emite una señal óptica infrarroja para ser detectada por el fotodiodo sin ser afectado ni por la luz solar ni por una fuente de luz artificial.

En la tabla 15 se presentan las principales características eléctricas del acondicionador de señal APDS-9700, destacando que es capaz de trabajar a una tensión de alimentación que va de 2,4V a 3,6V, que aunque no es una tensión que podamos obtener directamente del ASIBOT no es un inconveniente, ya que se ha solucionado mediante un convertidor cc/cc de 5V a 3,3V.

Tabla 15. Características eléctricas APDS-9700

Recommended Operating Conditions

Supply Voltage	V _{CC}	2.4	3.6	V
----------------	-----------------	-----	-----	---

Electrical & Optical Specifications (Ta=25°C)					
Parameters	Symbol	Minimum	Typical	Maximum	Units
Input					
Logic High Voltage, LEDON	V _{IH}	1.6		V _{CC}	V
Logic High Voltage, ENB	V _{IH}	1.4		V _{CC}	V
		1.5		V _{CC}	V
		1.7		V _{CC}	V
Logic Low Voltage, LEDON	V _{IL}	0		0.3	V
Logic Low Voltage, ENB	V _{IL}	0		0.3	V

Este acondicionador de señal presenta diversas entradas y salidas a través de sus pines y mediante los cuales se puede configurar el dispositivo. En la figura 45 se puede ver el pinout del componente.

APDS-9700 pin-out and I/O Configurations

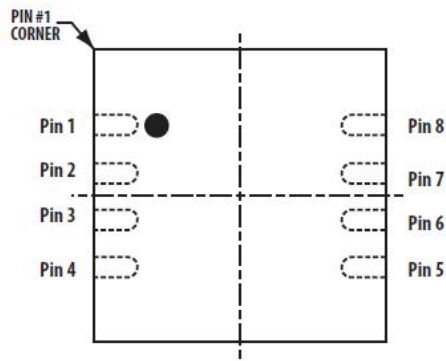


Figura 45. Pinout APDS-9700

- Pin 1 (LED ON): Pin de entrada digital por el que introducimos la señal PWM que genera los pulsos de luz infrarroja emitidos por el emisor.
- Pin 2 (ENB): Pin de entrada digital Enable. Con un bit controlaremos la activación o desactivación del APDS-9700. Puede ser desactivado para maximizar el ahorro de energía, y por tanto el consumo de batería en caso de que sea utilizado en una aplicación portátil.
ENB = 0 Normal mode operation / ENB = 1 Shut down mode
- Pin 3 (DOUT): Pin de salida digital. Proporciona un pulso a nivel alto si se detectase un objeto. Da flexibilidad al usuario dándole la posibilidad de obtener

una respuesta digital de la detección de un objeto si la aplicación así lo requiere.

$DOUT = 0$ when $VPFILT > VTH$ / $DOUT = Vcc$ when $VPFILT < VTH$

- Pin 4 (GND): Tierra
- Pin 5 (PD): Pin de entrada analógica. Se conecta al catodo del fotodiodo y por este pin el APDS-9700 recibe la señal analógica que genera el sensor óptico al detectar un objeto.
- Pin 6 (PFLIT): Pin de salida analógica. Por este pin obtenemos la señal analógica, generada por el sensor óptico, tratada y preparada para ser utilizada.
- Pin 7 (LEDA): Pin de Salida analógica. Está conectado al ánodo del led infrarrojo. Por este pin se obtiene una señal PWM que polariza el led emisor y lo protege contra sobre tensión.
- Pin 8 (VCC): Voltaje de alimentación.

3.2.3 El microcontrolador

Las principales características del microcontrolador PIC18F2580 seleccionado están resumidas en la tabla 16 en el que cabe destacar el módulo ECAN , el modulo CCP y el convertidor ADC de 10 bits de resolución con 8 canales de entrada, que a continuación se describirán más detalladamente. Además este microcontrolador incluye 4 *TIMER*, de los cuáles uno será utilizado para controlar ciertas características temporales de los Nodos CANopen. Posee una memoria de programa de 32K, que permite hasta un máximo de 16384 instrucciones simples de programa.

Tabla 16. Principales características del microcontrolador

CARACTERISTICAS	PIC18F2580
I/O Ports	Ports A, B, C, (E) Ports
Timers	4
Capture/Compare/PWM Modules	1
ECAN Module	1
10-bit Analog-to-Digital Module	8 Input Channels

3.2.3.1 Módulo controlador CAN

Las características que posee el módulo ECAN del microcontrolador son las siguientes:

- Permite una tasa de transferencia de hasta 1Mbps.
- Cumple la normativa 2.0B según las especificaciones dadas por Bosch.
- Tres modos de operación (Legacy (0), Enhanced Legacy (1), FIFO (2)).
- Tres buffers dedicados a transmisión.
- Dos buffers dedicados a recepción.
- Seis buffers programables para transmisión ó recepción.

- Tres máscaras de recepción de 29 bits.
- Dieciséis filtros de recepción de 29 bits asociables dinámicamente.
- Tratamiento automático de Tramas Remotas.
- Características de tratamiento de error avanzado.
- Longitud de datos de 0-8 bytes.
- Tasa de bits programable hasta 1Mbit/seg.
- Fuente de reloj programable
- Modo de bajo consumo (Sleep Mode).

3.2.3.2 Módulo CCP

El módulo CCP (Captura/Comparación/PWM) es un periférico interno que permite medir y controlar diferentes eventos.

El modo de captura proporciona el acceso al estado actual de un registro que cambia su valor constantemente. Para el modelo 18F2580, son los registros de los temporizadores Timer1 y Timer3.

El modo de comparación compara constantemente valores de dos registros. Uno de ellos es el registro del temporizador Timer1. Este circuito también le permite al usuario activar un evento externo después de que haya expirado una cantidad de tiempo predeterminada.

PWM (Pulse Width Modulation - modulación por ancho de pulsos) puede generar señales de frecuencia y de ciclo de trabajo variados por uno o más pines de salida.

El microcontrolador PIC18F2580 dispone de un módulo CCP (CCP1), que para este diseño se utilizará para generar la PWM necesaria para polarizar el led infrarrojo emisor y así emitir los pulsos de luz infrarroja que se encargaran de la detección de objetos. La configuración de este puerto se detalla en el apartado 3.3.2 de este documento.

3.2.3.3 Convertidor ADC

El modulo convertidor analógico digital del PIC 18F2580 tiene ocho entradas. Este modulo permite convertir una señal de entrada analógica en una señal digital de 10 bits.

Cada puerto asociado al convertidor A/D puede ser configurado como una entrada analógica o como entradas y salidas digitales.

El modulo convertidor cuenta con cinco registros:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)

- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

Los registros ADRESH y ADRESL contienen el resultado de la conversión A/D. Cuando la conversión A/D se completa el resultado se carga en estos registros. El registro ADCON0 controla el funcionamiento del modulo A/D. El registro ADCON1 configura las funciones de los pines del puerto. Y el registro ADCON2 configura el reloj fuente del A/D, el tiempo de adquisición programada y la justificación.

Una vez que el modulo de conversión A/D ha sido configurado como se desea, el canal de conversión debe ser indicado antes de que esta comience. Los canales de entrada analógica deben tener sus TRISbits seleccionados como entradas. Una vez configurado el tiempo de adquisición y este haya transcurrido, comienza la conversión. El tiempo de adquisición puede reconfigurarse entre el reset del bit GO/DONE producido a la finalización de la conversión y la siguiente conversión.

Los pasos para realizar una conversión A/D son los siguientes:

1. Configurar el modulo A/D:
 - Configurar pines analógicos, tensión de referencia y E/S digitales. (ADCON1)
 - Selección el canal de entrada del A/D.(ADCON0)
 - Seleccione el tiempo de adquisición.(ADCON2)
 - Seleccione el reloj del convertidor A/D (ADCON2)
 - Activar un modulo A/D.(ADCON0)
2. Configurar interrupción del A/D si se desea:
 - Reset del bit ADIF
 - Set del bit ADIE
 - Ser del bit GIE
3. Esperar el tiempo de adquisición requerido si es necesario.
4. Inicio de conversión:
 - Set del bit GO/DONE.(ADCON0)
5. Espere que la conversión A/D se complete, mediante:
 - Reset del bit GO/DONE
 - Esperando la interrupción del A/D
6. Lea los registros de resultados (ADRESH; ADRESL)
 - Reset del bit ADIF si es necesario.
 -

3.2.4 El transmisor-receptor

Este transceiver seleccionado ha sido el MCP2551, que tiene como principales características soportar la tasa de transferencia máxima de 1 Mbps que se especifica

en el protocolo CANopen y permite conectar un máximo de 112 nodos al Bus. En la figura 46 se puede observar el pinout del componente, cuyo patillaje y funcionalidad de los mismos se muestran en la tabla 17.

PDIP/SOIC

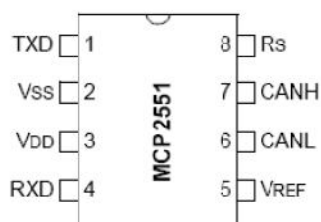


Figura 46. Pinout del transceiver

Tabla 17. Función de los pines del transceiver

Pin Number	Pin Name	Pin Function
1	TXD	Transmit Data Input
2	VSS	Ground
3	VDD	Supply Voltage
4	RXD	Receive Data Output
5	VREF	Reference Output Voltage
6	CANL	CAN Low-Level Voltage I/O
7	CANH	CAN High-Level Voltage I/O
8	Rs	Slope-Control Input

La descripción detallada de cada pin se muestra a continuación:

- Pin 1: TxD: Entrada digital que recibe los datos que van a ser enviados por el Bus.
- Pin 2: VSS: Referencia de tierra.
- Pin 3: VDD: Tensión de alimentación (+5V).
- Pin 4: RxD: Señal digital que transmite los datos recibidos por el bus.
- Pin 5: Vref: Tensión de referencia a la salida definida como $V_{dd}/2$.
- Pin 6: CANL: Parte baja de la señal diferencial del Bus.
- Pin 7: CANH: Parte alta de la señal diferencial del Bus.
- Pin 8: RS: Este pin permite seleccionar distintos modos de operación (High Speed, Slope Control, Standby) mediante una resistencia externa.

El transceiver es capaz de operar en tres modos, dependiendo de la tasa de transferencia deseada y el nivel de emisiones EMI (Emisiones ElectroMagnéticas) permitidas en el sistema. La tasa máxima de transferencia es controlada por medio de una resistencia externa (RS) que permite modificar los tiempos de la pendiente de transición (slew rate) de las señales CANH y CANL. Mientras que las emisiones EMI son controladas por dicha pendiente y mediante un regulador interno de las señales CANH y CANL que proporciona un control simétrico entre ambas señales.

Los modos de operación son los siguientes:

- High Speed: Las pendientes de transición son lo más rápidas posibles para conseguir tasas de transferencias muy altas. Este modo se selecciona

conectando directamente el pin RS a masa. Este es el método que utilizamos para nuestro sistema ya que necesitamos la máxima tasa de transferencia posible.

- Slope Control: Reduce aun más las emisiones EMI, controlando el slew rate de las señales CANH y CANL, a costa de limitar la velocidad máxima de transferencia. La pendiente de la señal será proporcional a la intensidad que circule por el pin RS. Este modo se selecciona conectando una resistencia (R_s) entre el pin RS y masa. En la figura 47 se ilustran los valores típicos de slew rate en función del valor de la resistencia externa.

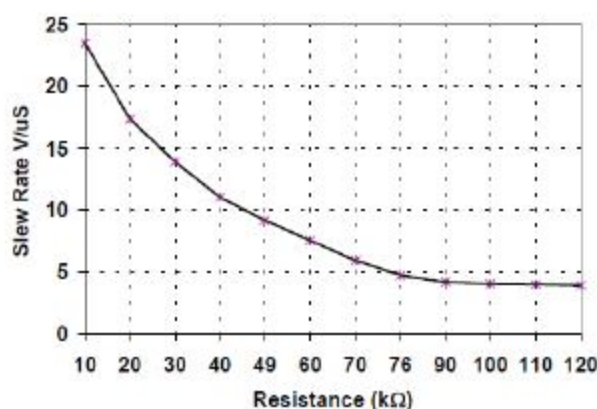


Figura 47 Valores de slew-rate en función de R_s

- Standby: Permite introducir al dispositivo en modo standby o sleep cuando el valor de R_s es muy elevado. En este modo se desactiva la transmisión, mientras que la recepción sigue operativa consumiendo muy poca intensidad. El microcontrolador puede seguir monitorizando el bus y cambiar al modo de operación normal cuando detecte actividad. El primer mensaje puede perderse cuando la tasa de transferencia es elevada.

3.2.5 El Multiplexor

El multiplexor seleccionado para el control de entrada de las señales analógicas al microcontrolador es el MC14551B, cuyo pinout podemos observar en la figura 48. Es un switch analógico controlado digitalmente que permite señales analógicas comprendidas en su rango de tensión de alimentación. Se trata de cuatro multiplexores 2 a 1 integrados en un mismo chip y que funcionan bajo el mismo bit de control. A continuación se define el patillaje y funcionalidad de los mismos.

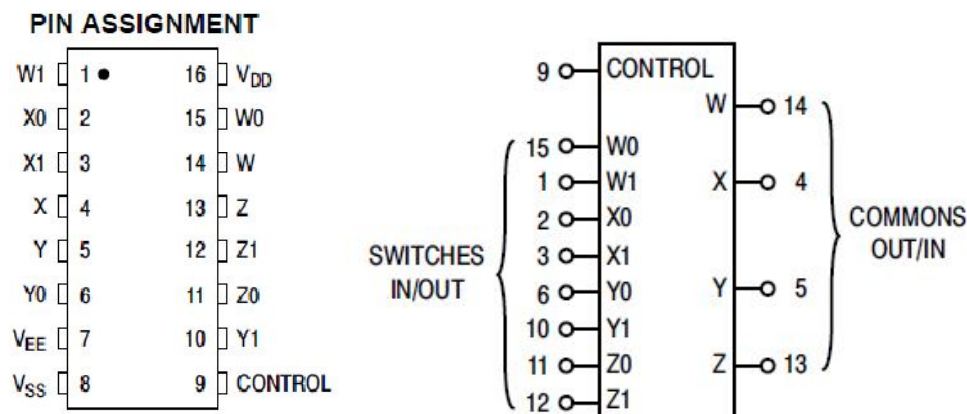


Figura 48. Pinout del multiplexor

En la tabla 18 podemos observar cómo se realiza la gestión de las señales a partir de bit de control

Tabla 18. Gestión de las señales

Control	ON
0	W0 X0 Y0 Z0
1	W1 X1 Y1 Z1

Como se puede observar en la figura 48 el multiplexor cuenta con 8 entradas analógicas que podemos gestionar mediante el pin de control.

En la tabla 19 se muestran las características eléctricas del MC14551B. Destacando el amplio rango de tensión de alimentación, que nos permite trabajar con todo tipo de señales analógicas que se encuentren en ese rango.

Tabla 19. Características eléctricas del multiplexor

MAXIMUM RATINGS (2.)

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage Range (Referenced to V_{EE} , $V_{SS} \geq V_{EE}$)	- 0.5 to + 18.0	V
V_{in} , V_{out}	Input or Output Voltage (DC or Transient) (Referenced to V_{SS} for Control Input & V_{EE} for Switch I/O)	- 0.5 to $V_{DD} + 0.5$	V
I_{in}	Input Current (DC or Transient), per Control Pin	± 10	mA
I_{sw}	Switch Through Current	± 25	mA

3.3 Diseño hardware.

La estructura del robot donde se vaya a montar el sistema anticolidión que se desarrolla en este proyecto es un factor importante a la hora de realizar un diseño hardware del mismo. Debido a que ASIBOT es un brazo robot, el sistema anticolidión debe adaptarse a su diseño. Se ha ideado que el sistema anticolidión tenga forma circular y que vaya localizado en la pinza, a modo de pulsera, de modo que durante el desplazamiento hacia un objeto concreto esta pulsera permita la detección de otros objetos cercanos con los que podría colisionar.

Una vez decido el aspecto físico de nuestro sistema anticolidión se han barajado diferentes opciones de cómo ubicar los sensores y cuantos serian suficientes para realizar una detección de objetos a una distancia y unos costes adecuados. En todo momento se tuvo en cuenta la necesidad de multiplexar las señales de salida de los sensores, ya que esto no permitía utilizar un microcontrolador más reducido en tamaño y coste.

Para evitar crear sectores de zona muerta se realiza la lectura de sensores de manera salteada, para ello es necesario el control de dos sensores contiguos por una misma salida de multiplexor, es decir, un multiplexor 2 a 1. Para obtener el máximo rendimiento de los multiplexores se pensó en múltiplos de 8 sensores para crear el anillo.

Los factores que determinara el número de sensores que formaran el anillo son: el haz de detección del sensor seleccionado y el coste que conlleva cada circuito sensor. El haz de detección del sensor lo podemos observar en la figura 49 y aproximadamente de 60° y el coste de cada circuito sensor lo podemos encontrar en el Anexo 2.

Para valorar la colocación de diferentes números de sensores se utilizo la herramienta Autocad. Se descarto la posibilidad de implementar el anillo sensor mediante 8 sensores, ya que la zona muerta que se generaba era lo suficiente como para que un objeto no fuera detectado y provocara una colisión indeseada. Las siguiente opciones barajadas fueron la de implementar el sistema anticolidión con 16 o 24 sensores. Se realizo una representación grafica, para ambas soluciones, mediante la herramienta Autocad y los resultados se muestran en las figuras 49 y 50 respectivamente.

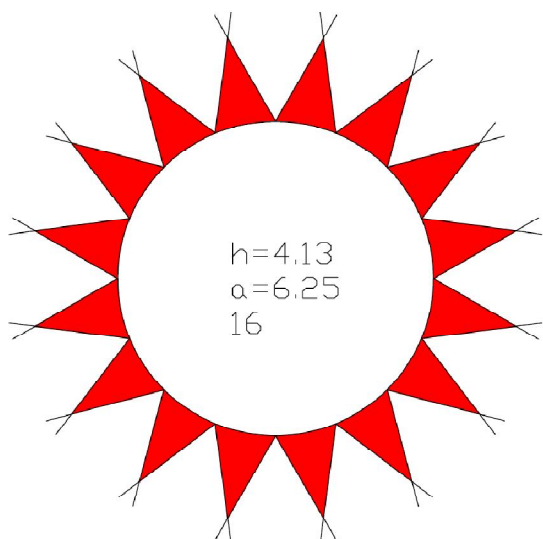


Figura 49. Anillo de 16 sensores

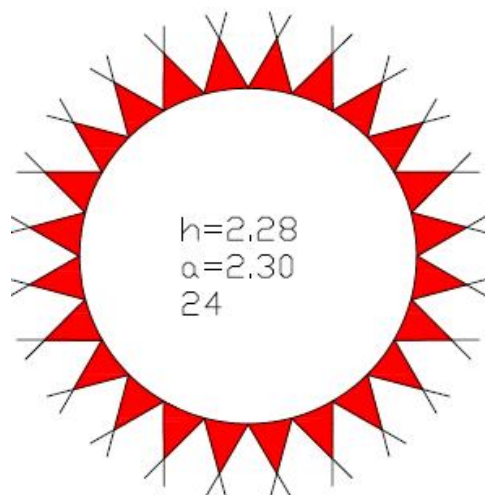


Figura 50. Anillo de 24 sensores

En las representaciones graficas anteriores se ha marcado en color rojo la zona muerta generada por la distribución de los circuitos sensores. Esta zona muerta tienen forma de triángulo y por ello se indica las dimensiones mediante la altura h y el área a .

Se puede observar que en la opción de 24 sensores la zona muerta generada es inferior a la opción de implementar el sistema anticollision con 16 sensores, sin embargo no solo debemos tener en cuenta la zona muerta que se genera en el sistema sino que debemos atender a las necesidades del diseño y uno de los objetivos es implementar un sistema anticollision de bajo coste e implementar la opción de 24 sensores supone un coste muy superior que la opción de 16 ya que no son solo 8 sensores mas sino que son 8 circuitos sensores compuestos por acondicionador de señal, sensor y una serie de componentes pasivos que permiten una correcta detección del objeto. Por ello se ha optado por la colocación de 16 sensores infrarrojos separados equitativamente a lo largo de la longitud de la pulsera. En la figura 49 podemos observar que la zona muerta que se genera en forma de triángulo con altura $h= 4,13\text{cm}$ y de área $a=6,25\text{cm}^2$. Es un área aceptable teniendo en cuenta que los sensores infrarrojos junto con un acondicionador de señal adecuado tienen un rango de detección desde prácticamente 5mm a 200mm, luego para que un objeto no fuera detectado, durante el movimiento del ASIBOT, este debería aparecer repentinamente en este área.

3.3.1 Circuito Sensor.

Debido a que el sistema anticolidión ira montado en ASIBOT que, como ya se ha explicado con anterioridad, se puede ubicar tanto en la silla de ruedas del usuario como en los conectores que le permiten desplazarse por los distintos puntos de la casa, es necesario acondicionar el sensor para que sea inmune a la luz solar y a la luz artificial.

En las características del sensor podemos ver que su rango de detección es de prácticamente 5mm a 60mm, lo cual hace que el sensor no sea capaz de detectar un objeto con la suficiente antelación como para evitar la colisión, luego es necesario un tratamiento para mejorar la distancia de detección.

Para el acondicionamiento del circuito sensor se ha utilizado el ya descrito acondicionador de señal APDS-9700, mediante el cual se pretende mejorar la robustez y eficiencia del sensor óptico.

Un sensor óptico como el utilizado en el circuito sensor (HDSL-9100), es un sensor de reflexión directa que consta de un emisor y un receptor en un mismo cuerpo, como ya se ha descrito. En este tipo de sensores el emisor convierte una señal eléctrica en una señal óptica y la transmite, indirectamente, al receptor o fotodiodo que se encarga de convertir una señal óptica en una señal eléctrica. La transmisión hacia el receptor es de manera indirecta ya que esta solo se produce cuando la señal óptica emitida es reflejada por un objeto hacia el receptor.

Como se describe en el apartado 3.2.1 el APDS-9700 , el acondicionador de señal se compone fundamentalmente de un amplificador de trans-impedancia (TIA), un circuito de cancelación luz del sol, un convertidor tensión / corriente (V-I) y un comparador de histéresis.

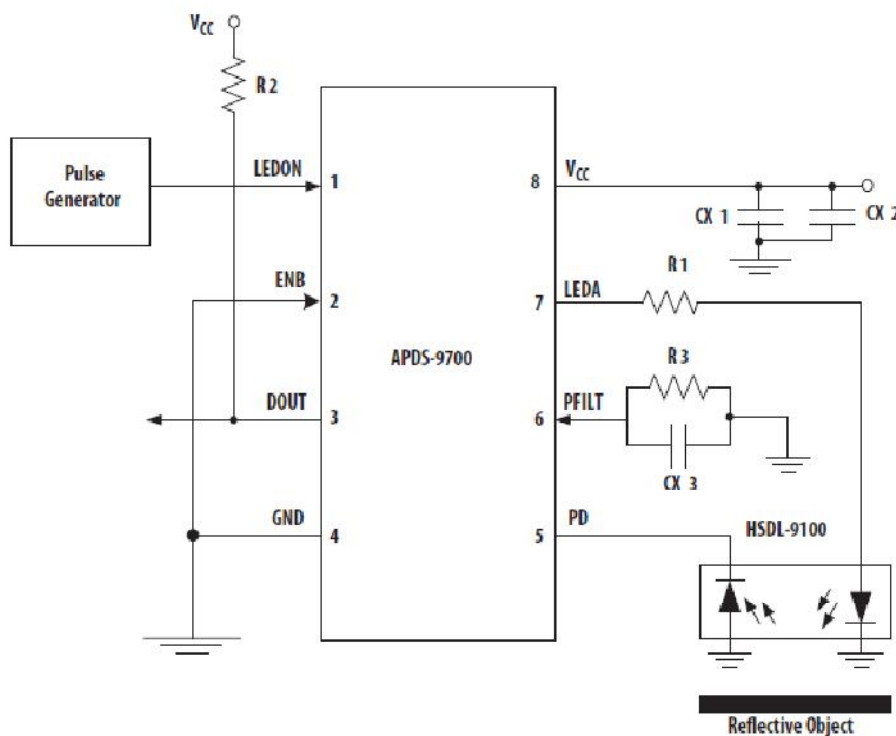


Figura 51. Esquema eléctrico del circuito sensor

La figura 51 lustra un diagrama del circuito sensor. En este circuito el emisor del sensor de proximidad irradia una secuencia de pulsos, que se transmiten por el medio, hasta que es reflejada por la superficie de un objeto a cierta distancia al fotodetector. Los pulsos son generados por un microcontrolador (18F2580) programado. Genera una PWM que es inyectada por el pin LEDON. Para un correcto funcionamiento del sensor HDSL-9100 la señal PWM tendrá un periodo de 100us, una amplitud aproximada de 3v y un ciclo de trabajo del 20% . CX1 y CX2 son condensadores de desacoplo para evitar picos que pueden provocar un mal funcionamiento del acondicionador de señal. LEDA es el pin de salida para polarizar el led emisor. R1 es una resistenciya que nos permite limitar la corriente que fluye por el emisor de infrarrojos. El cátodo del fotodetector se conecta directamente a al pin PD con respecto a tierra. R3 y CX3 están conectados en paralelo con respecto a tierra en el pin PFLIT, y actúan como integrador para producir un voltaje de salida. La corriente interna del convertido V-I carga el condensador y se descarga a través de la resistenciya con una constante de tiempo particular. Los valores de la resistenciya R3 y CX3 condensador garantizan un bajo consumo de energía y un buena sensibilidad en la detección de distancia. El pin PFLIT se conecta a un puerto convertidor A/D del microcontrolador para que sea tratada. El pin de Enable no permite apagar el acondicionador de señal con un '1' lógico o encenderlo con un '0' lógico, esto nos permitiría ahorros de energía, en casos necesarios.

En la figura 52 podemos observar las distintas versiones V0 y V1, del circuito sensor, que se han utilizado para validar el diseño. La versión V0 esta implementada con

componentes de encapsulado DIP, mientras que la versión V1 esta implementada mediante componentes con encapsulado SMD. En la figura 53 también podemos observar la versión final V3, del diseño, cuya única diferencia con la versión V2 es la colocación de un conector, que oblige la recolocación de los distintos componentes que lo forman, para una más fácil sustitución del circuito sensor en caso de mal funcionamiento.

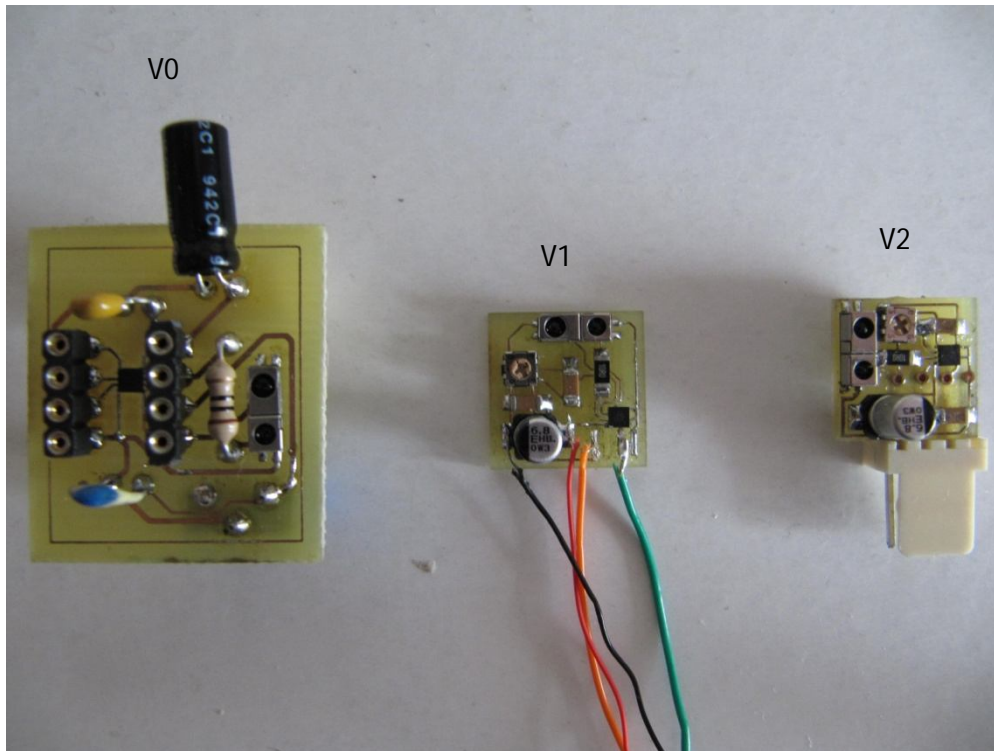


Figura 52. Versiones del circuito sensor

3.3.2 Circuito Control

El circuito de control está formado por el circuito de programación, que se explica en el apartado 3.2.3 de esta memoria, y todo lo necesario para realizar el control de los 16 sensores que formaran el anillo anticolidión, multiplexores que gestionan la entrada de señales al microcontrolador, componentes pasivos para configuración, led indicadores de estado y transceiver para la comunicación con el ASIBOT. En la figura 53 podemos observar un esquema eléctrico aproximado del circuito de control.

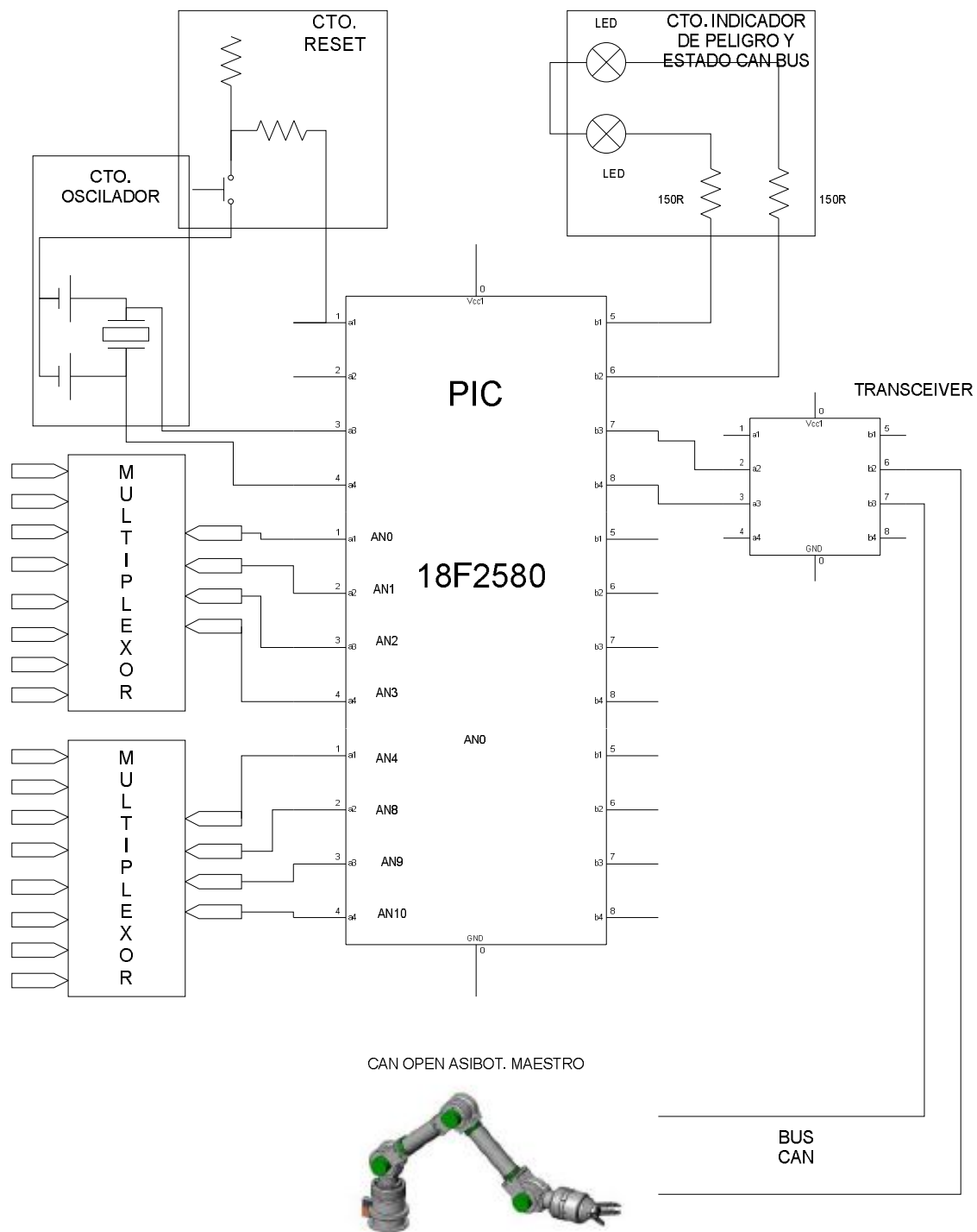


Figura 53. Esquema circuito de control

Se ha creado una placa PCB (Printed Circuit Board, placa de circuito impreso). Para el Layout de la placa se han tenido en cuenta la configuración de los distintos dispositivos para que se cumplan las especificaciones del diseño, y finalmente se ha obtenido una placa como la que se puede observar en la figura 54.

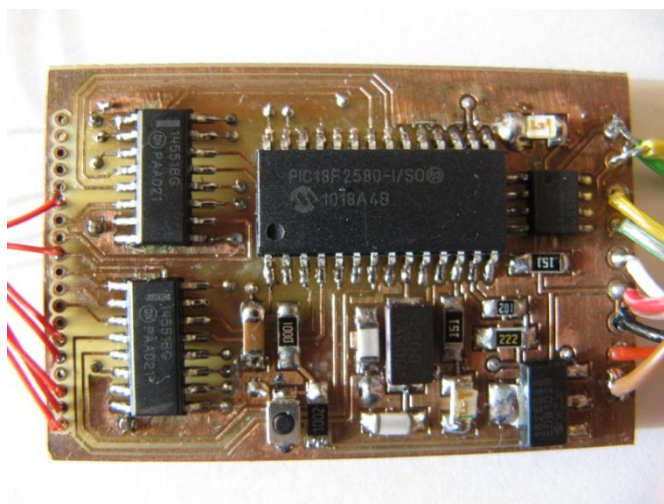


Figura 54. Circuito de control

En el circuito de control se puede ver implementado un circuito de reset, formado por una resistencia de $100\ \Omega$, un pulsador referenciado a la masa del circuito y una resistencia de pull-up de $1\ k\Omega$. Un cristal de cuarzo de 20MHz con sus respectivos condensadores de desacoplo de 22pF, valores seleccionados conforme a las recomendaciones del fabricante del microcontrolador que nos permitirá generar la señal PWM adecuada para excitar nuestro led infrarrojo emisor. Dos diodos leds, uno para indicar el estado del nodo de comunicaciones CAN y otro indicador de que nos encontramos en una distancia de peligro de colisión. Un conector RJ-11 de 6 pines, del que se utilizaran 5 para implementar el circuito de programación del microcontrolador que se describirá en el apartado 3.2.3. Las señales del microcontrolador necesarias para la programación de éste serán PGC (27) y PGD (28). Las señales CANRX (24) y CANTX (23) se conectaran a las patillas RxD y TxD del transceiver MCP2551 respectivamente.

Las conexiones realizadas en el transceiver MCP2551 serán las siguientes:

- Las señales RXD (4) y TXD (1) según se ha definido anteriormente.
- El pin RS (8) se conectara a masa para seleccionar el modo High Speed.
- Las señales CANL (6) y CANH (7) se conectaran al conector DB9 que permitirá la conexión con otros dispositivos CANopen. En la tabla 20 se muestra el conexionado del conector para la comunicación CAN.

Tabla 20. Conexiones para comunicación CAN

DB9	SEÑAL
2	CAN_L
7	CAN_H

En el circuito de de control se han colocado 2 multiplexores MC14551B para poder gestionar las señales analógicas que vienen de los 16 sensores que conforman el anillo anticolidión. Como ya se vio en el apartado 3.1.5 los multiplexores se controla mediante un bit (CONTROL). El bit de CONTROL de cada uno de los multiplexores están unidos entre sí y son gestionados mediante software por el pin 16 del microcontrolador (RC5). En la tabla 18 podíamos ver como es la selección de entradas y salidas de nuestros multiplexores a través de su bit de control. El multiplexor MC14551B tiene 8 entradas y 4 salidas de manera que la señal de salida de 8 sensores estarán conectadas a las 8 entradas de uno de los multiplexores y la salida del los 8 sensores restantes estarán conectadas al otro multiplexor. Estas conexiones no se realizan aleatoriamente, sino que para no crear zonas muertas a la hora de gestionar las señales de los sensores pares se conectaran en las entradas X0, Y0, W0 y Z0 del multiplexor 1 y a las entradas X0, Y0, W0 y Z0 del multiplexor 2. De la misma manera las señales de los sensores impares se conectaran en las entradas X1, Y1, W1 y Z1 del multiplexor 1 y a las entradas X1, Y1, W1 y Z1 del multiplexor 2.

3.3.3 Circuito de Programación.

Con el fin de poder programar el microcontrolador PIC sin necesidad de retirarlo (o desoldarlo de la PCB) y utilizar otros circuitos auxiliares se ha decidido introducir un conector micro-macth de 6 pines que permite tener acceso a las patillas de programación del microntrolador. A este modo de programación se le conoce como programación ICSP (InCircuit Serial Programming).

Programar los microcontroladores sin tener que desmontarlos de los circuitos donde están ubicados es una particularidad que tienen casi todos los dispositivos, pero si, además, su memoria de programa es del tipo Flash les confiere aún mayor flexibilidad, pues se pueden realizar cambios en los programas y volver a reprogramar el microcontrolador tantas veces como sea necesario para depurar la aplicación. A la hora de llevar a cabo este tipo de programación/depuración del microcontrolador se usará la herramienta que proporciona el entorno de desarrollo de Microchip, MPLAB ICD 2.

En la figura 55 se observa el conector de programación con los pines del PIC, a través de los cuales se produce la reprogramación del microcontrolador.

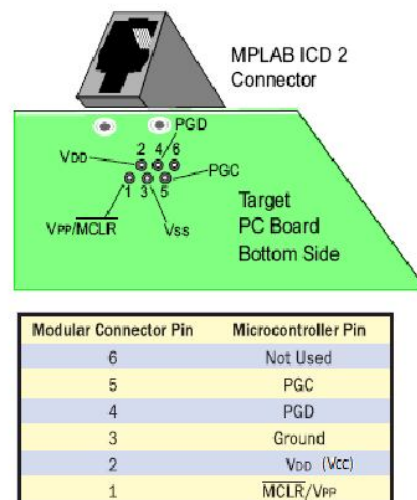


Figura 55. Conexión para la programación del microcontrolador

Aunque se dispone de seis pines solo será necesario conectar cinco de éstos. Una resistencia de pull-up ($1K\Omega$) conectada a la señal MCLR, que se utiliza como reset del PIC. Las señales VDD y VSS se conectan a VCC y GND del microcontrolador respectivamente, y por último las señales PGD y PGC. No todos los microcontroladores poseen las señales AVDD y AVSS pero en caso de tenerlas, deben ser conectadas como indica MPLAB ICD 2.

Existen unas directrices que proporciona MPLAB para que este modo de operación actúe correctamente. Son las enumeradas a continuación:

- No colocar resistencias de pull-up en PGC y PGD, ya que estas líneas poseen unas resistencias de pull-down de $4,7 K\Omega$ dentro del depurador ICD 2.
- No colocar condensadores en PGC y PGD, ya que estos evitarían rápidas transiciones en las señales de datos y de reloj durante las operaciones de programación y depuración.
- No colocar condensadores en MCLR, ya que estos evitarían transiciones rápidas a Vpp. Como ya se ha comentado, una resistencia es suficiente.
- No colocar diodos entre PGC y PGD, ya que provocarían que no hubiera comunicación bidireccional entre el dispositivo y el depurador.

Estas especificaciones se pueden observar en la figura 56.

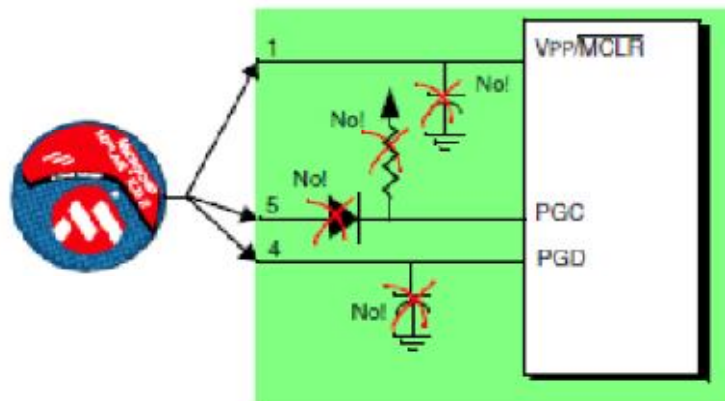


Figura 56. Directrices para la correcta programación

3.3.4 Esquemático final

A continuación, en la figura 57 se muestra una representación en bloques del esquema final del sistema. En el podemos observar que el sistema anticollision lo forman n circuitos sensores y un circuito de control tal y como se ha descrito durante este capítulo.

Se diseñará la placa de circuito impreso con ORCAD Layout, teniendo en cuenta los tamaños de los componentes, con el fin de conseguir una placa de tamaño mínimo para que el diseño final sea lo más compacto posible. El esquemático se añadirá a los anexos.

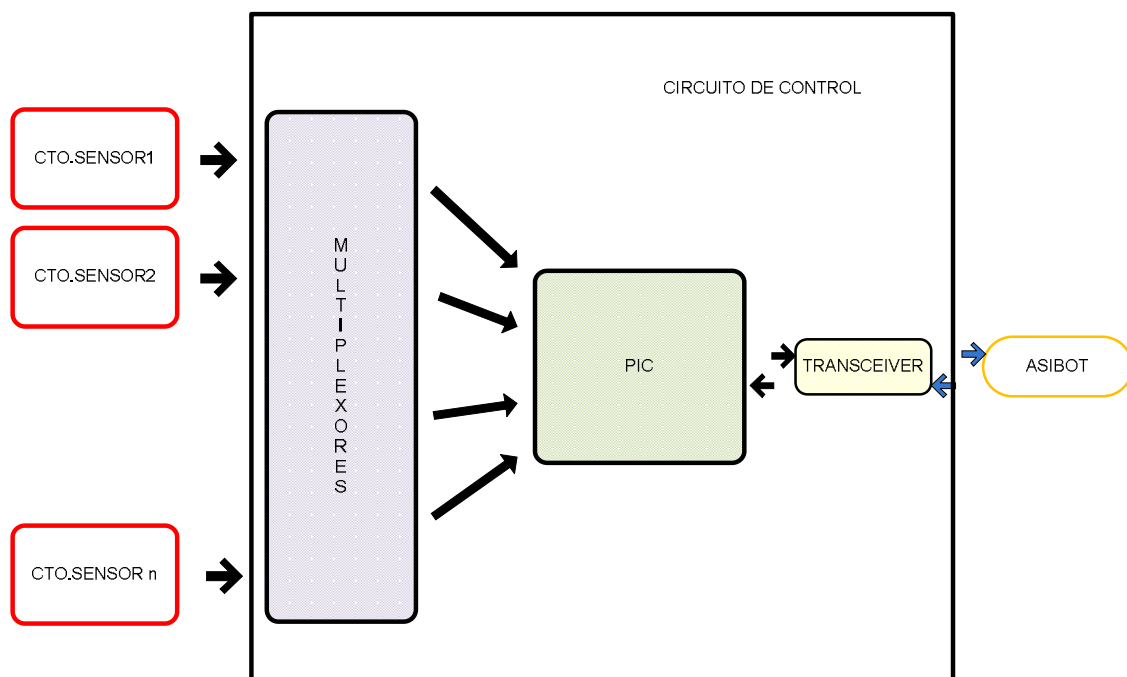


Figura 57. Esquemático del sistema anticollision

En la figura 58 podremos observar el sistema anticolisión implementado simplemente con 4 circuitos sensores.

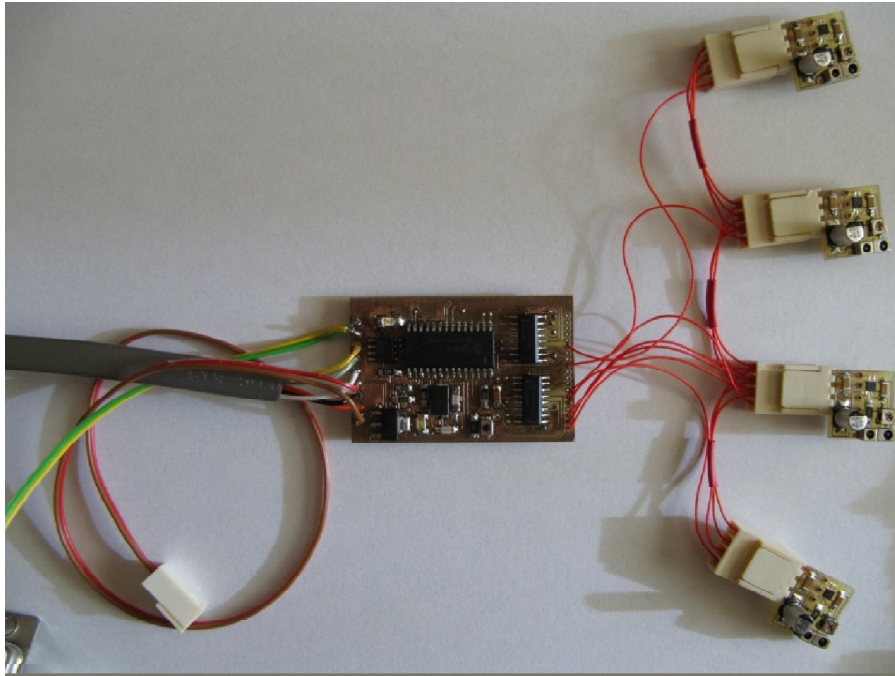


Figura58. Sistema anticolisión

En el esquema se pueden diferenciar los dos circuitos explicados en apartados anteriores y sus diagramas de bloques podemos verlos en la figura 59 y 60:

- Circuito Sensor:

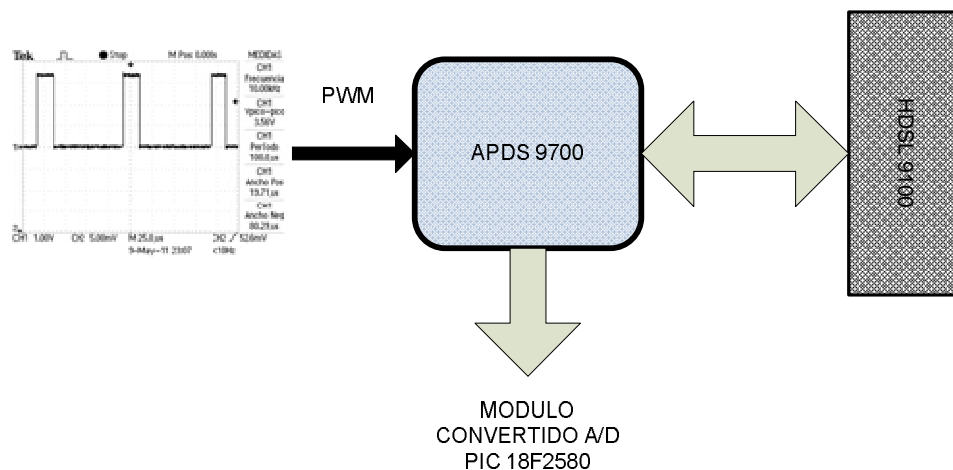


Figura 59. Esquema Bloques del circuito sensor

El circuito sensor recibe la señal PWM, generada por el microcontrolador, a través del acondicionador de señal, que realiza un filtrado tanto de la luz artificial como de la natural. Los sensores se encargan de detectar al objeto y proporciona una señal analógica, en función de la distancia, que el acondicionador se encarga de amplificar y dar salida.

- Circuito de Control:

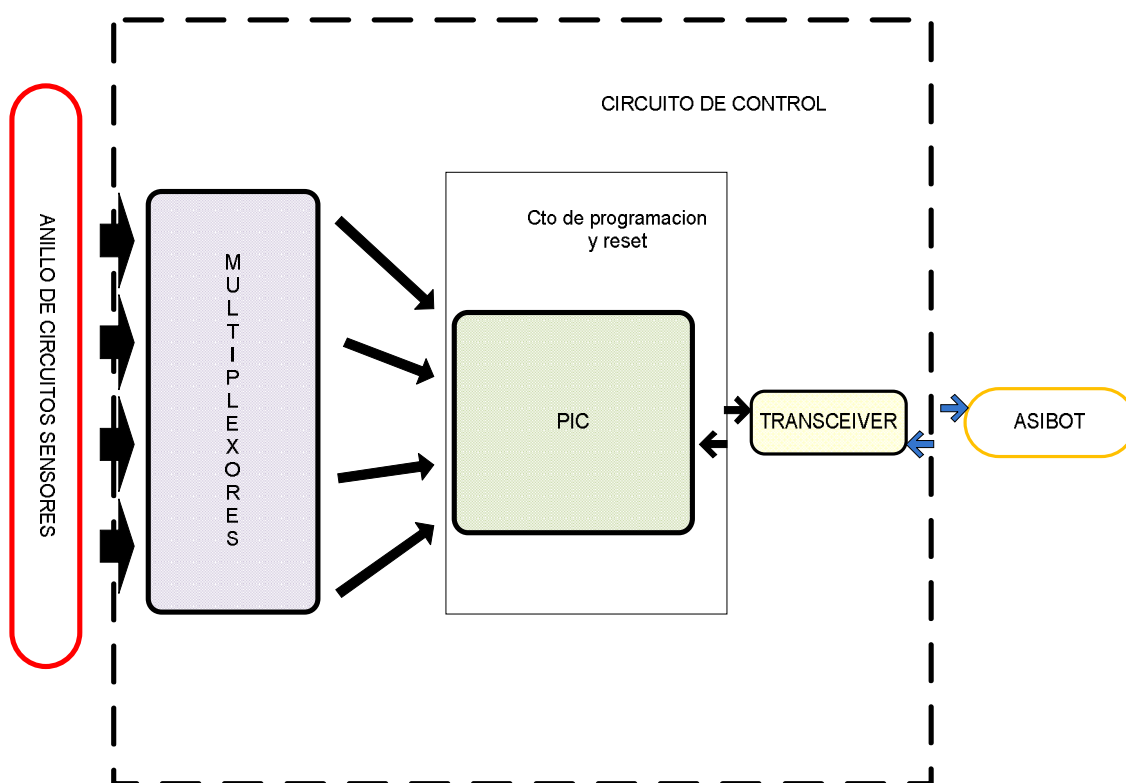


Figura 60. Esquema en Orcad del circuito de programación

El punto de conexión entre el circuito sensor y el de control es el acondicionador de señal que envía la señal recibida desde el sensor, amplificada, a los multiplexores que se encargan de gestionar la entrada de las señales de los sensores al microcontrolador. El microcontrolador se encarga de tratar y procesar la información recibida desde los sensores y genera una trama de datos para informar al ASIBOT de la distancia del objeto. La trama es transmitida al ASIBOT mediante el transceiver.

En las figuras 61 y 62 se puede observar el layout de los dos circuitos que forman el sistema anticollisión:

- Circuito sensor:
La placa de circuito impreso del circuito sensor tiene unas dimensiones de 14mm x 14mm. Este circuito se alimenta a 3.3V recibidos desde el circuito de control y tiene un consumo aproximado de 0.1A

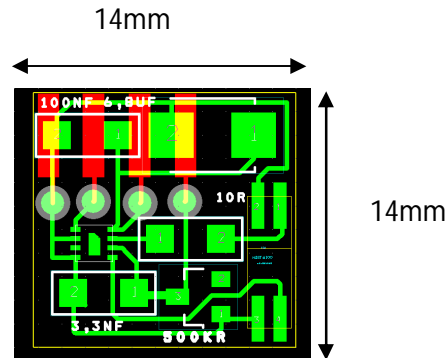


Figura 61. Layout del circuito sensor

- Circuito de control:

La placa de circuito impreso del circuito sensor tiene unas dimensiones de 29,8mm x 44,6 mm. Este circuito se alimenta a 5V recibidos desde cualquiera de las articulaciones de ASIBOT.

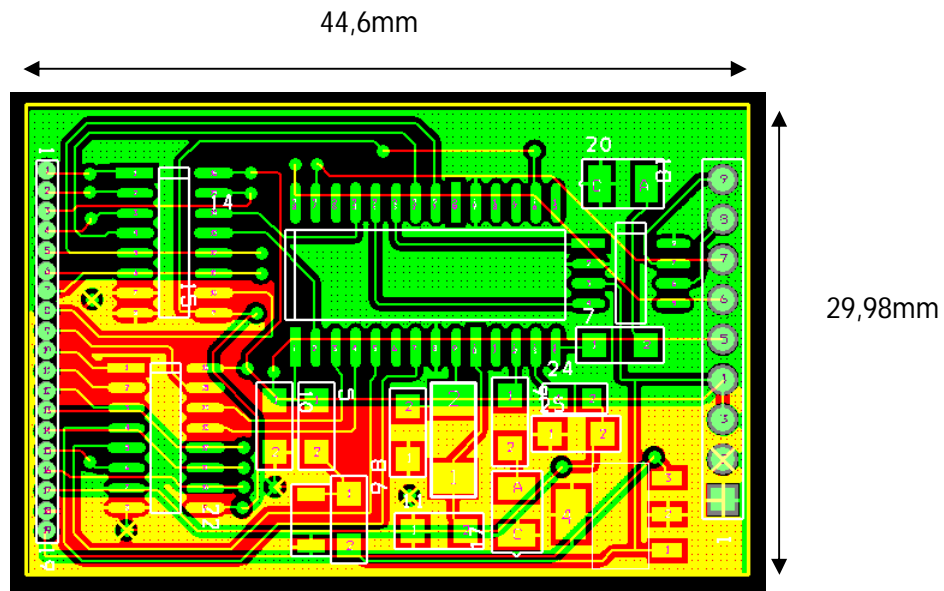


Figura 62. Layout del circuito de control

3.4 Diseño software

El desarrollo software del proyecto comprende toda la programación necesaria del microcontrolador PIC. Esta parte junto con el desarrollo hardware componen en general la estructura principal del proyecto.

El software se ha desarrollado en lenguaje C utilizando el entorno de programación MPLAB y el compilador C18, ambos de Microchip®. Se ha utilizado este lenguaje ya que es muy potente para la programación de microcontroladores y permite hacer más legible el programa, además de simplifica la tarea del programador con respecto a las instrucciones de ensamblador.

El programa se ha desarrollado de forma estructurada permitiendo la modificación del mismo con facilidad y basándose en la Application Note 945 (AN945 "A CANopen Stack for PIC18 ECAN Microcontrollers"). La pila de Microchip AN945 es una librería de archivos C que implementan un Nodo CANopen de Entradas/Salidas y que junto con la documentación del protocolo CANopen (DS- 301) dan soporte para implementar cualquier dispositivo que trabaje bajo dicho protocolo.

El archivo ApplR.c contiene la rutina principal del programa, a través del cual se configuran todos los parámetros iniciales y posteriormente se ejecutan todos los procesos relacionados con la aplicación y comunicación. A continuación en la figura 63 se puede ver el flujograma del programa.

3.4.1 Configuración software del microcontrolador

```

:
#pragma config OSC = HS ,FCMEN = OFF,IESO = OFF //CONFIG1H (HS porque cristal superior a 3.5 MHz)
#pragma config PWRT = ON,BOREN = BOHW,BORV = 3 //CONFIG2L
#pragma config WDT = OFF,WDTPS = 32768 //CONFIG2H
#pragma config MCLRE = ON,LPT1OSC = OFF,PBADEN = OFF//CONFIG3H
#pragma config STVREN = ON,LVP = OFF,XINST = OFF,DEBUG = ON //CONFIG4L
#pragma config CP0 = OFF,CP1 = OFF,CP2 = OFF//CONFIG5L
#pragma config CPB = OFF,CPD = OFF//CONFIG5H
#pragma config WRT0 = OFF,WRT1 = OFF,WRT2 = OFF//CONFIG6L
#pragma config WRTB = OFF,WRTC = OFF,WRTD = OFF//CONFIG6H
#pragma config EBTR0 = OFF,EBTR1 = OFF,EBTR2 = OFF,EBTR3 = OFF//CONFIG7L
#pragma config EBTRB = OFF//CONFIG7H
#pragma config BBSIZ = 1024 // 1K words (2K bytes) Boot Block
:

```

A través de estas líneas se llevan a cabo las instrucciones necesarias para una correcta configuración del microcontrolador 18F2580, en las que se resalta como más importante:

- OSC = HS, mediante la cual se selecciona la velocidad del oscilador, en este caso, de alta velocidad ya que el cristal conectado es de 20 MHz.
- WDT=OFF, el timer del perro guardián (watch dog) desactivado para así poder utilizar el método de programación on-chip.
- MCLRE=ON, para indicar que existe un circuito de reset externo para restablecer todos los valores iniciales del micro en caso de ser necesario.

El resto de instrucciones se utilizan para proteger código, bloques de arranque y de registros contra lectura y escritura.

Una vez configurado el microcontrolador en sí, se procede a la configuración de los distintos pines de éste que se van a utilizar en este diseño.

```

:
TRISCbits.TRISC3 = 0; //el pin RC3 como salida LED INDICADOR STOP
TRISCbits.TRISC5 = 0; //el pin RC5 como salida CONTROL MULTIPLEXORES
TRISCbits.TRISC2 = 0; //el pin CCP1 como salida PARA LA PWM
TRISCbits.TRISC1 = 0; //el pin RC1 como salida LED INDICADOR ESTADO CAN
TRISAbits.TRISA0 = 1; // pin AN0 como entrada al ADC
TRISAbits.TRISA1 = 1; // pin AN1 como entrada al ADC
TRISAbits.TRISA2 = 1; // pin AN2 como entrada al ADC
TRISAbits.TRISA3 = 1; // pin AN3 como entrada al ADC
TRISAbits.TRISA5 = 1; // pin AN4 como entrada al ADC
TRISBbits.TRISB1 = 1; // pin AN8 como entrada al ADC
TRISBbits.TRISB4 = 1; // pin AN9 como entrada al ADC
TRISBbits.TRISB0 = 1; // pin AN10 como entrada al ADC
:

```

RC1 y RC3 se han utilizado para encender leds y por tanto como pines de salida, que nos indicaran el estado en el que se encuentra el modulo CAN y si existe peligro de colisión respectivamente. RC5 se ha configurado como pin de salida para realizar el control de los dos multiplexores que gestionan las señales analógicas de entrada en el puerto ADC de 8 canales que tiene el microcontrolador. RC2 se configura como pin de salida para la señal PWM necesaria para el diodo led infrarrojo emisor. Todos los pines correspondientes a los 8 canales analógicos del convertidor A/D se configuran como entrada, ya que será por estos pines por los que se llevara a cabo la comunicación entre nuestros sensores y el microcontrolador.

3.4.2 Configuración software de la señal PWM

En la tabla 21 se muestran la funciones, contenidas en la librería pwm.h, compatibles con el periférico PWM y una breve descripción de su aplicación.

Tabla 21. Funciones contenidas en la librería pwm.h

FUNCION	DESCRIPCION
ClosePWMx	Deshabilita el canal PWM
OpenPWMx	Configura el canal PWM
SetDCPWMx	Establece el valor del ciclo de trabajo
SetOutputPWMx	Establece los bits de configuración de salida de la PWM para ECCP
CloseEPWMx	Deshabilita el canal Enance PWM
OpenEPWMx	Configura el canal Enance PWM
SetDCEPWMx	Establece el valor del ciclo de trabajo
SetOutputEPWMx	Establece los bits de configuración de salida de la EPWM para ECCP

Con base a los diferentes registros de control, los bits de configuración y sus posiciones en el registro de control hay que saber la versión del dispositivo a configurar, en nuestro caso PIC18F2580, para llamar a la función apropiada con el número correcto de argumentos.

A continuación se muestra la tabla 22 en la que aparece la versión PWM para nuestro dispositivo:

Tabla 22. Versión de pwm compatible con el microcontrolador

VERSION	DISPOSITIVOS COMPATIBLES
PWM_V1	18F2580,

De la tabla anterior las funciones más importantes para nuestro diseño y por tanto las que se usaran para la configuración de la señal PWM son las siguientes:

- ClosePWMx
- OpenPWMx
- SetDCPWMx

Close PWM1

PROTOTIPO:

```
void ClosePWM1 (void);
```

Se trata de una función que nos permite deshabilitar el canal específico de PWM, si es necesario. Está contenida en la librería pwm.h que debemos añadir en la cabecera de nuestro programa.

Es una función tipo void sin argumento. La sentencia en nuestro programa con la que deshabilitaríamos el canal PWM es la siguiente:

```
ClosePWM1;
```

OpenPWM1

PROTOTIPO:

```
void OpenPWM1( char period);
```

Es una función que también está incluida en la misma librería que la función ClosePWM1. Mediante esta función podemos configurar el canal PWM. Nos permitirá establecer el periodo de la señal.

El periodo puede ser cualquier valor entre 0x00 y 0xFF. Este valor determina la frecuencia de la PWM utilizando la siguiente fórmula:

$$\text{PWM periodo} = [(\text{periodo}) + 1] \times 4 \times \text{TOSC} \times \text{TMR2 prescaler}$$

Esta función configura el canal de PWM para un periodo especificado y como base de tiempos de la PWM utiliza solo el Timer2 del microcontrolador. Esto significa que además de abrir el canal de PWM el Timer2 también debe abrirse, antes de que la PWM este operativa.

Para configurar la PWM debemos realizar la llamada a la función e inicializar el argumento del periodo. Al tratarse de una función tipo void, y una vez calculado el periodo de la señal PWM la sentencia será la siguiente:

```
OpenPWM1 (0xFF);
```

La apertura del Timer2 se realiza mediante la siguiente sentencia:

```
OpenTimer2();
```

`void OpenTimer2(unsigned char config);` es el prototipo de la función que nos permite configurar el timer2. Se encuentra dentro de la librería `timer.h`, que debemos incluir en la cabecera del programa.

El argumento `config` de la función, es una máscara de bits que se crea mediante la realización, bit a bit, de cualquiera de las operaciones `AND('&')` o `OR('|')`, configurable por el usuario, el valor de cada una de las categorías se muestran a continuación.

Enable Timer2 Interrupt:	<code>TIMER_INT_OFF</code>	Interrupt disabled
Prescale Value:	<code>T2_PS_1_16</code>	1:16 prescale
Postscale Value:	<code>T2_POST_1_1</code>	1:1 postscale

SetDCPWM1

PROTOTIPO:

```
void SetDCPWM1( unsigned int dutycycle );
```

Utilizamos esta función para establecer el valor del ciclo de trabajo de la señal PWM. El valor del ciclo de trabajo puede ser cualquier número de 10bits. Solo los 10 bits de menor peso se escribirán en el registro del ciclo de trabajo. El ciclo de trabajo o más específicamente la forma de onda de la PWM se calcula mediante la siguiente fórmula:

$$PWM \times Duty \ cycle = (DCx<9:0>) \times TOSC$$

Donde `DCx<9:0>` es el valor de 10 bits que se especifica en la llamada a la función.

La máxima resolución de la forma de onda de la PWM puede ser calculada mediante la siguiente fórmula:

$$Resolution \ (bits) = \log(FOSC/F_{pwm}) / \log(2)$$

Para realizar la configuración debemos realizar la llamada a la función e inicializar el argumento del ciclo de trabajo. Al tratarse de una función tipo `void` y una vez calculado el ciclo de trabajo de la señal PWM la sentencia será la siguiente:

```
SetDCPWM1(0);
```

Los pasos a seguir para realizar la configuración de una señal PWM son los siguientes:

1. Incluir en la cabecera del programa las librerías que contienen las funciones
2. Establecer como pin de salida el puerto CCP
3. Abrir el Timer2

4. Abrir el canal PWM y configurar el periodo deseado
5. Configurar el ciclo de trabajo de la PWM

A continuación podremos observar los fragmentos de programa que configuran la señal PWM, tal y como se indica en los pasos anteriores:

```

1. #include <pwm.h> #include<timer.h>
2. TRISCbits.TRISC2 = 0; //el pin CCP1 como salida PARA LA PWM
3. OpenTimer2(TIMER_INT_OFF & T2_PS_1_16 & T2_POST_1_1);
4. OpenPWM1(0x1d); // configuramos período/Fpwm 10kHz
5. SetDCPWM1(24); // elegimos ciclo de trabajo de la señal PWM al 20%

```

3.4.3 Configuración software de puertos ADCs y conversión de datos.

En la siguiente tabla 23 se muestran la funciones, contenidas en la librería adc.h, compatibles con el periférico convertidor A/D y una breve descripción de su aplicación.

Tabla 23. Funciones contenidas en la librería adc.h

FUNCION	DESCRIPCION
BusyADC	Detecta si el modulo A/D está realizando una conversión
CloseADC	Deshabilita el convertidor A/D
ConvertADC	Empieza la conversión
OpenADC	Configura el convertidor A/D
ReadADC	Lee el resultado de la conversión
SetChanADC	Selecciona el canal que se va a utilizar
SelChanConvADC	Selecciona el canal que se va a utilizar y empieza la conversión

Con base a los diferentes registros de control, los bits de configuración y sus posiciones en el registro de control hay que saber la versión del dispositivo a configurar, en nuestro caso PIC18F2580, para llamar a la función apropiada con el número correcto de argumentos.

A continuación, en la tabla 34, se muestra la versión ADC para nuestro dispositivo:

Tabla 24. Versión de adc compatible con el microcontrolador

VERSION	DISPOSITIVOS COMPATIBLES
ADC_V5	18F2580,

De la tabla 23, las funciones más importantes para nuestro diseño y por tanto las que se usaran para la configuración el convertidor A/D son las siguientes:

- OpenADC
- SetChanADC
- ConvertADC
- BusyADC
- ReadADC

OpenADC

PROTOTIPO:

```
void OpenADC(unsigned char config, unsigned char config2, unsigned char portconfig);
```

Esta función nos permite resetear el periférico A/D a un estado inicial y configura los registros relativos al convertidor A/D mediante los argumentos config y config2.

Tanto el argumento config, como el argumento config2, de la función, son una máscara de bits que se crea mediante la realización, bit a bit, de cualquiera de las operaciones AND('&') o OR('|'), configurable por el usuario. El valor de cada una de las categorías definen la frecuencia del reloj utilizado por el convertido, la resolución del resultado de la conversión, tiempo de adquisición del señal a convertir, habilitación o deshabilitación de interrupciones, voltajes de referencia y tipo de señal que se van a recibir.

Config

A/D clock source: Define la frecuencia del reloj utilizado por el convertidor

```
ADC_FOSC_2  FOSC / 2
```

A/D result justification: Define la resolución del resultado de la conversión

```
ADC_RIGHT_JUST  Result in Least Significant bits
```

A/D acquisition time select: Este parámetro define el número de adquisiciones/muestras a tomar de la señal a convertir.

```
ADC_12_TAD  12 Tad
```

Config2

Channel:

ADC_V5 and ADC_V6:

ADC_CH0 Channel 0

A/D Interrupts: Permite habilitar o deshabilitar las interrupciones

ADC_INT_OFF Interrupts disabled

A/D Vref+ and Vref- configuration: Define los voltajes de referencia para el resultado digital de la conversión

ADC_REF_VDD_VSS

VREF+ = VDD & VREF- = VSS

Portconfig

ADC_V5 and ADC_V6: Define el tipo de entrada que se recibirán por los puertos ANx

ADC_15ANA All analog

SetChanADC

PROTOTIPO:

```
void SetChanADC(unsigned char channel);
```

Esta función nos permite seleccionar el canal de entrada al convertidor A/D.

Channel es un argumento configurable por el usuario, con el canal que se desea seleccionar. La elección de debe realizar con un valor de la categoría enumerada a continuación.

Channel: Nos permite seleccionar el canal de entrada del convertidor

ADC_CH0	Channel 0
ADC_CH1	Channel 1
ADC_CH2	Channel 2
ADC_CH3	Channel 3
ADC_CH4	Channel 4
ADC_CH8	Channel 8
ADC_CH9	Channel 9
ADC_CH10	Channel 10

ConvertADC

PROTOTIPO:

```
void ConvertADC(void);
```

Esta función nos permite, mediante su llamada, dar orden para que comience la conversión.

BusyADC

PROTOTIPO:

```
char BusyADC(void);
```

Mediante esta función se hace la comprobación de si el convertidor A/D está realizando una conversión o no. La función nos devuelve un carácter indicador.

‘1’ si el periférico A/D está realizando una conversión

‘0’ si no la está realizando

ReadADC

PROTOTIPO:

```
int ReadADC(void);
```

Esta función nos permite leer el resultado de la conversión. Devuelve el resultado de 10-bits de la conversión A/D.

Los pasos a seguir para realizar una conversión A/D son los siguientes:

1. Incluir en la cabecera del programa la librería que contiene las funciones compatibles con el modulo convertidor A/D
2. Configurar los pines analógicos como entradas
3. Configurar los registros relativos al convertidor A/D
4. Seleccionar el canal de entrada de la señal analógica a convertir
5. Comenzar la conversión
6. Comprobar que la conversión a finalizado
7. Devolver el resultado de la conversión.

A continuación podremos observar los fragmentos de programa que configuran la el convertidor A/D, tal y como se indica en los pasos anteriores:

```
#include <adc.h>
...
unsigned int conversion (unsigned char canal);
...
1. TRISAbits.TRISA0 = 1; // pin AN0 como entrada al ADC
   TRISAbits.TRISA1 = 1; // pin AN1 como entrada al ADC
   TRISAbits.TRISA2 = 1; // pin AN2 como entrada al ADC
```

```

    TRISAbits.TRISA3 = 1; // pin AN3 como entrada al ADC
    TRISAbits.TRISA5 = 1; // pin AN4 como entrada al ADC
    TRISBbits.TRISB1 = 1; // pin AN8 como entrada al ADC
    TRISBbits.TRISB4 = 1; // pin AN9 como entrada al
ADC
    TRISBbits.TRISB0 = 1; // pin AN10 como entrada al ADC
    ...

    2.  OpenADC (ADC_FOSC_2 & ADC_RIGHT_JUST & ADC_12_TAD,ADC_CH0 & ADC_REF_VDD_VSS &
        ADC_INT_OFF,ADC_15ANA ); //configura los registros para la conversión A/D
    ...
    unsigned int conversion (unsigned char canal)
    {
        3.  SetChanADC (canal); // selecciona el canal
        Delay10TCY x(2); // Retardo, porque después de cada conversión necesita un retardo antes de comenzar la
        siguiente
        4.  ConvertADC(); // Comienza la conversión
        5.  while (BusyADC ()); // Espera a terminar la conversión
        6.  return ReadADC(); // Devuelve el resultado de la conversión
    }

```

Como podemos observar en los fragmentos de código anteriores, para la conversión A/D se ha creado una función llamada conversión, que consta de un argumento mediante el que seleccionamos el canal que se utilizara para la siguiente conversión. Al utilizar una función específica para la conversión, a los pasos de cómo se configurar y realiza una conversión A/D hemos añadido la declaración y llamada de la función en cuestión.

3.4.4 Rutina de adquisición de datos.

Una vez configurada la señal PWM y el modulo convertidor A/D, es necesario crear una rutina de comunicación entre nuestro circuito de control y los circuitos sensores que componen el anillo anticollision. A continuación se explican los pasos para el almacenamiento y adquisición de los datos que proporcionan los circuitos sensores.

En primer lugar necesitamos crear una variable donde almacenar la distancia a la que se encuentra el objeto detectado. Debido a que nuestro anillo de sensores está formado por 16 circuitos sensores, se define un array de 16 posiciones en las que se volcara el resultado digitalizado de las señales recibidas desde los sensores.

```

unsigned int distanciarR [15]; //array de distancias de los 16 sensores, 0 a 7 impares , 8 a 15 pares

```

Para digitalizar la señal entregada por cada sensor, es necesario realizar una conversión A/D. Esta se realiza mediante la llamada a la función conversión, que tiene como argumento el canal por el que entrara la señal analógica a convertir. Para la elección de canales, se ha definido un array de 8 posiciones, una para cada canal, y se

ha inicializado con unos caracteres predeterminados correspondientes a la nomenclatura adecuada para la selección de canales.

```
unsigned char channel [] = {ADC_CH0, ADC_CH1, ADC_CH2, ADC_CH3 ,ADC_CH4, ADC_CH8, ADC_CH9, ADC_CH10}; //array de constantes para seleccionar los canales para conversión
```

Tanto el array de variables que almacenara el resultado de la distancia al objeto, como el array para los canales, se han creado para que nos proporcionen cierta agilidad a la hora de realizar barridos de detección.

Para una detección de objetos adecuada se ha programado un barrido constante, primero de los sensores impares y luego de los pares, con el fin de no crear zonas muertas alrededor de la pinza de ASIBOT.

Como ya se ha explicado con anterioridad el convertidor A/D consta de 8 canales de entrada y por tanto es necesaria la multiplexación de señales para una correcta comunicación entre los circuitos sensores y el microcontrolador. A continuación se muestra y explica el fragmento de programa que se encarga de programar los barridos de detección y la multiplexación de las señales.

```

:
PORTCbits.RC5 = 1; //chequeamos sensores impares
for (i=0; i<8; i++)
    for (j=0; j<8; j++)
    {
        distancia[i]=conversion(channel[j]);
        if (distancia[i]>62)
            PORTCbits.RC3 = 1; //enciende led
            uIRDistance.word=distanciaIR[i];
        else
            PORTCbits.RC3 = 0; //led apagado
    }

PORTCbits.RC5 = 0; //chequeamos sensores pares
for (i=8; i<16; i++)
    for (j=0; j<8; j++)
    {
        distancia[i]=conversion(channel[j]);
        if (distancia[i]>62)
            PORTCbits.RC3 = 1; //enciende led
            uIRDistance.word=distanciaIR[i];
        else
            PORTCbits.RC3 = 0; //led apagado    }
:

```

Se utiliza el pin 5 del puerto C para gestionar el control sobre los multiplexores. En la primera parte de la rutina de adquisición este pin se configura con un '1' lógico para leer los sensores impares. En la segunda parte se configura con un '0' lógico para leer los sensores pares.

Se utiliza dos sentencias repetitivas anidadas, como es el for, que recorren los array definidos para el almacenamiento de las distancias y la elección de los canales en la llamada a la función. En la primera parte se almacenan en las primeras 8 posiciones de array distancia los resultados de las conversiones analógico-digital que se han realizado a través de los canales del conversor A/D. En la segunda parte los resultados se almacenan en las ultimas 8 posiciones del array.

Cada vez que se almacena un dato se compara con la distancia de peligro, configurable por el usuario, para que en caso de que exista peligro de colisión emita una señal luminosa indicativa a través de un led y transmita dicha distancia a la CPU de control de ASIBOT, por CANbus, para que este actúe en consecuencia a la situación que se plantea.

3.5 Configuración software de la comunicación CAN

Para un correcto funcionamiento de nuestro sistema anticolidión debe existir una comunicación con ASIBOT, que como ya se ha comentado se realizara por CANbus.

A continuación en la figura 64 vemos un diagrama de comunicación entre el controlador host y el esclavo CAN.

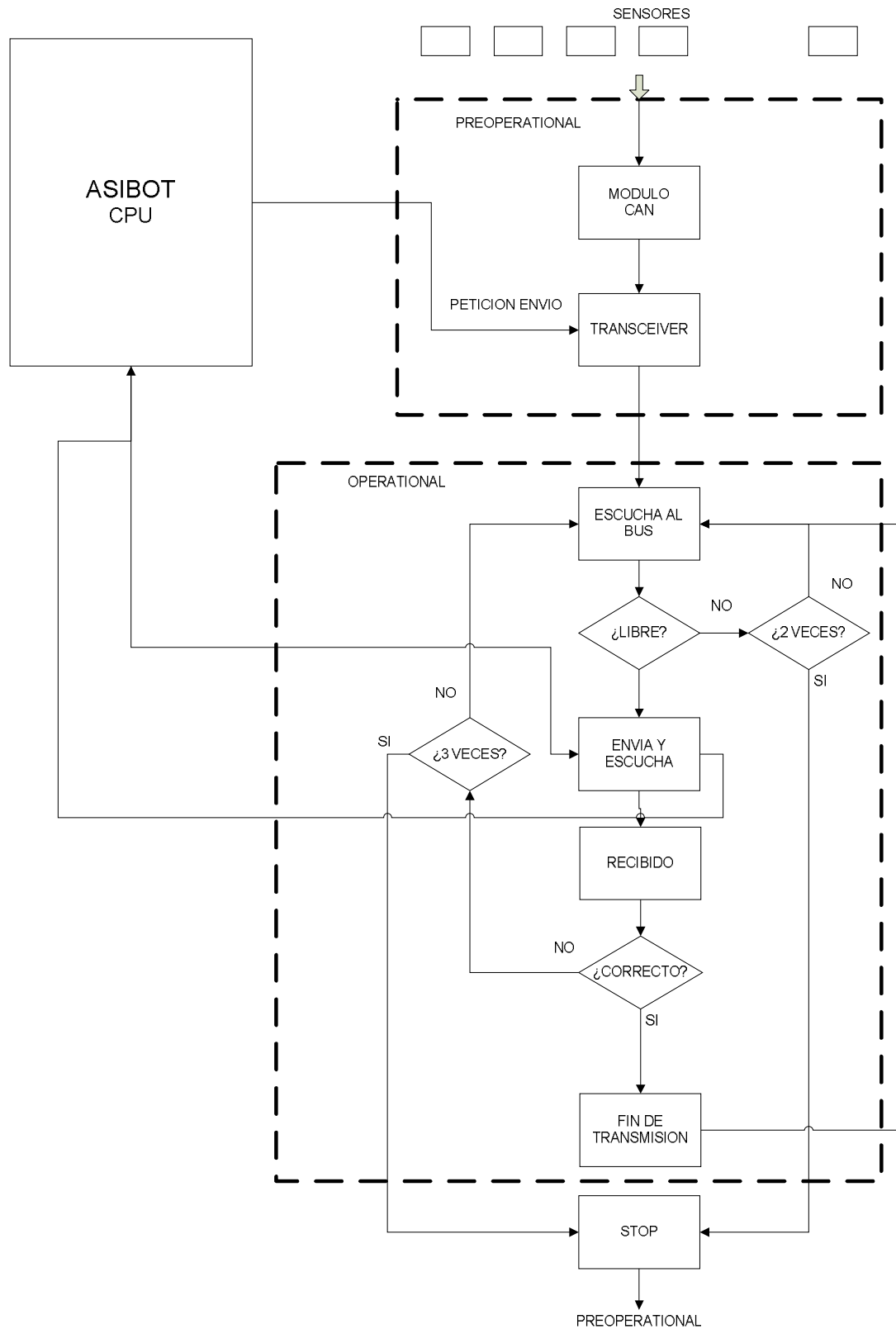


Figura 64. Diagrama de actividad

Para realizar la configuración software de esta comunicación es necesario conocer las características de la pila CANopen que proporciona Microchip® para poder realizar la implementación deseada.

Para ello primero se hará una descripción de la pila y sus características para, a continuación, desarrollar el código y las aplicaciones necesarias para integrar el módulo diseñado en una red CAN.

3.5.1 Descripción de la pila CANopen de Microchip®

La pila CANopen de Microchip® no es simplemente un enfoque genérico de una pequeña aplicación, sino que esta es una pila de comunicación genérica basada en CANopen que puede ser modificada según las necesidades del usuario. Está basada en la documentación estándar DS-301 de CAN in Automation (CiA)[13]. De hecho, la mayoría del código está limitado a implementar áreas de las especificaciones dadas por CiA, con especial énfasis en dar a conocer al usuario como debe desarrollar una aplicación bajo el protocolo CANopen. Esta pila implementa el desarrollo de una aplicación basada en el perfil de dispositivo DS-401, es decir, un módulo de I/O genérico.

Todo el código suministrado con esta Application Note está desarrollado para los PIC18F8680 y PIC18F4680, los cuales incluyen tecnología ECAN, y se recomienda que sean compilados con la versión V2.30 (ó superior) del compilador C18 de Microchip®. Aunque el código esta desarrollado para los microcontroladores mencionados anteriormente, esta pila es fácilmente adaptable a otros microcontroladores de la familia PIC18 que incluya tecnología CAN.

3.5.2 Características principales de la pila CANopen

La pila CANopen implementa las capas bajas del protocolo y permite implementar sobre esta la capa de aplicaciones. Esta pila está dividida en una serie de archivos fuentes y de cabecera más pequeños, escrita todo en C y que permite a los usuarios seleccionar los servicios necesarios y selectivamente construir un proyecto adaptado a las especificaciones de su diseño. Aun así, la aplicación y ciertos aspectos de la comunicación deben ser desarrollados por el usuario. En la tabla 25 se muestra una lista completa de los archivos fuente.

Tabla 25. Ficheros de la pila CANopen

Nombre archivo	Descripción
CO_CANDRV.c CO_CANDRV.h	Driver de configuración del modulo ECAN. Pueden ser sustituidos por los de otros dispositivos si es necesario
CO_COMM.c CO_COMM.h	Servicios de gestión de las comunicaciones. Requerido para todas las aplicaciones
CO_DEV.c CO_DEV.h	Archivos específicos de cada dispositivo. Deben ser editados por el usuario
CO_DICT.c CO_DICT.h CO_DICT.def	Estructura del diccionario de objetos. Requeridos para todas las aplicaciones
CO_MAIN.c CO_MAIN.h	Servicios principales de CANopen. Requeridos para todas las aplicaciones
CO_MEMIO.c CO_MEMIO.h	Funciones de copiado de memoria usadas por el diccionario. Requeridos por todas las aplicaciones
CO_NMT.c CO_NMT.h	Gestión de los endpoint de comunicaciones del bus
CO_NMTE.c CO_NMTE.h	Endpoint de comunicación para los servicios Node Guard, Heartbeat y Boot-up
CO_PDO.c CO_PDO.h	Servicios PDO generales
CO_PDO1.c CO_PDO1.h CO_PDO2.c CO_PDO2.h CO_PDO3.c CO_PDO3.h CO_PDO4.c CO_PDO4.h	Endpoint de tratamiento de objetos PDO. Proporciona una plantilla que requiere ser desarrollada por el usuario para cada aplicación en concreto. Deben ser utilizados con los archivos de servicios PDO generales
CO_SDO1.c CO_SDO1.h	Endpoint de comunicaciones SDO que debe implementarse por defecto
CO_SYNC.c CO_SYNC.h	Endpoint de comunicaciones para los consumidores del objeto SYNC
CO_TOOLS.c CO_TOOLS.h	Herramientas de conversión de identificadores entre el formato microchip y CANopen. Todos los COB-ID están guardados en formato microchip y son convertidos a formato CANopen cuando es necesario
CO_ABERR.h	Definiciones de errores comunes. Requerido para todas las aplicaciones

Las características más destacables en este diseño incluyen:

- Máquina de estados encargada de manejar todas las comunicaciones entre los Nodos y Objetos.
- Un Service Data Object
- Hasta 4 transmit y 4 receive Process Data Objects (TPDOs y RPDOs).
- Soporte para transferencia de mensajes acelerados y segmentados.
- Soporte de mapeado de PDOs estático.
- Estructura para el Diccionario de Objetos PDO y SDO.

- Node Guard/Life Guard.
- Consumidor de Objetos SYNC.
- Productor de Heartbeat.
- Soporte para la configuración del módulo ECAN.

3.5.3 Uso de la pila CANopen

En este apartado se hará una descripción más detallada de los archivos de la pila CANopen que han sido modificados para adaptarse al diseño de sistemas CANopen basados en microcontroladores PIC18 de Microchip®.

El archivo main.c contiene la rutina del programa principal, a través del cual se configuran todos los parámetros iniciales y posteriormente se ejecutan todos los procesos relacionados con la aplicación y comunicación.

En cuanto a la configuración, sigue los siguientes pasos:

1. Configura, llamando a la función `void TimerInit(void)`, el timer_0 para generar un evento cada 1 ms que será utilizado para controlar todos los procesos temporales del sistema.
2. Almacena el COB-ID de sincronismo, en formato CANopen, mediante la llamada a la función `mSYNC_SetCOBID(SYNC_COB)`, donde SYNC_COB representa el identificador de SYNC. Posteriormente el COB-ID de sincronismo es transformado a formato Microchip mediante la llamada `mTOOLS_CO2MCHP(mSYNC_GetCOBID())` y almacenado en memoria tras invocar de nuevo a la función `mSYNC_SetCOBID(mTOOLS_GetCOBID())`. La función `mTOOLS_GetCOBID()` devuelve el valor del COB-ID de sincronismo en formato Microchip.
3. Asigna el NodeID del dispositivo a través de la función `mCO_SetNodeID(NodeID)`.
4. Selecciona la tasa de transferencia del Bus por medio de la función `mCO_SetBaud(bitrate)`, donde bitrate es un valor perteneciente al rango [0...8] y que estará asociado a los parámetros de configuración de la tasa de transferencia.
5. Configura el Node Guard ó Heartbeat por medio de las funciones `mNMTE_SetHeartBeat(HeartBeat)`, `mNMTE_SetGuardTime(GuardTime)`, `mNMTE_SetLifeFactor(LifeFactor)`, donde HeartBeat y GuardTime representan tiempos dados en milisegundos y LifeFactor es un factor utilizado en el Node Guard.
6. Llamada a la función de inicialización de todos los parámetros relacionados directamente con la aplicación que se desea desarrollar, en este caso la función `void IRIInit(void)`.

7. Inicializar el Nodo. Esta inicialización se realiza a través de la llamada `mCO_InitAll()`, que realiza otra sub-llamada a la función `void _CO_COMMResetEventManager(void)` donde se resetea el estado del Nodo y se producen sucesivas llamadas a otras funciones para configurar todo el hardware relacionado con la tasa de transferencia (`mCANReset(CANBitRate)`) y con la implementación de los endpoint tales como el NMT (`void_CO_COMM_NMT_Open(void)`), el SYNC consumer (`void_CO_COMM_SYNC_Open(void)`), el SDO por defecto (`void _CO_COMM_SDO1_Open(void)`) y el Node Guard ó Heartbeat (`void _CO_COMM_NMTE_Open(void)`). Una vez creados todos los endpoint se asigna los valores de estos a los filtros de los buffers de entrada (`void _CANEventManager(void)`), se activa el modo Normal de operación del módulo CAN (`mCANOpenComm()`) y por último se activa el modo PREOPERATIONAL después de enviar el mensaje de Boot-up.

La ejecución de los procesos asociados a la aplicación y comunicaciones se realizan a través del código incrustado dentro de un bucle infinito while de la función main.

En este apartado existen tres instrucciones claramente diferenciadas, que son las siguientes:

1. Procesado de todos los eventos CANopen relacionados con las comunicaciones a través de la función `mCO_ProcessAllEvents()`, tales como PDOs, SDOs, SYNC consumer, etc.
2. Procesado de las aplicaciones realizadas por el Nodo. Esta función ha de ser desarrollada por el usuario y ha de realizar todas las funciones específicas de la aplicación que se desea desarrollar. En este caso la función se llama `void IRProcessEvents (void)` y se encuentra en el archivo `AppIR.c`.
3. Tratamiento de todos los procesos relacionados con eventos temporales tales como Node Guard ó Heartbeat, control de tiempo de caducidad de transferencia SDO segmentada, tiempos de inhibición de los PDOs, etc. Este apartado se encuentra implementado en la función `mCO_ProcessAllTimeEvents()` y se ejecuta cada vez que el `timer_0` se desborda. Es decir, cada vez que la función `unsigned charTimerIsOverflowEvent (void)` devuelve el valor TRUE.

3.5.3.1 Funciones configuradas

`void IRInit (void)`

Es una función desarrollada por el usuario y que configura los pines de E/S, los parámetros del microcontrolador relacionados con la lectura del sensor como el modulo convertidor A/D e inicializa el Timer2 necesario para la generación de la PWM. Esta función se encuentra localizada en el archivo `AppIR.c`.

El siguiente paso que realiza la función es configurar los modos de sincronismo de los TPDOs según el perfil del dispositivo. Se han de habilitar dos PDOs, uno en modo asíncrono (TPDO_1) y el otro en modo síncrono (TPDO_2). Es decir, el PDO_1 transmitirá un mensaje cada vez que detecte un cambio de posición y el PDO_2 enviara el mensaje de posición de forma sincronizada con el SYNC Object.

Esta configuración se logra a través de las variables `uIRSyncSet_1`, `uIRSyncCount_2` y `uIRSyncSet_2`. `uIRSyncSet_1` configurara el TPDO_1 y las otras dos variables el TPDO_2 adoptando los siguientes valores. `uIRSyncSet_1 = 254`, configura el TPDO_1 para transmisión asíncrona. `uIRSyncCount_2 = uIRSyncSet_2 = 1`, configura el TPDO_2 para transmisión acíclica síncrona.

El siguiente paso es inicializar los buffers de transmisión de ambos TPDOs (`uLocalTPDO2Buffer` y `uLocalTPDO1Buffer`). En estos buffers se almacenaran los datos de posición que serán enviados según el modo de sincronización de cada TPDO.

Por último, en esta función se han de crear los COB-ID de los PDOs, asociarlos al buffer de transmisión e indicar el tamaño de dicho buffer. A continuación se muestra el proceso para el PDO_1 ya que para los siguientes PDOs el código no varía.

```
⋮  
//Crear el identificador y convertirlo en formato Microchip  
mTOOLS_CO2MCHP(mCOMM_GetNodeID().byte + 0xC0000180L);  
// Almacenar el COB-ID del TPDO1  
mTPDOSetCOB(1, mTOOLS_GetCOBID());  
// Asociar buffer del PDO1  
mTPDOSetTxPtr(1, (unsigned char *)&uLocalTPDO1Buffer[0]);  
⋮
```

void IRProcessEvents (void)

Es la función principal que se encarga de realizar la lectura de los sensores, el control de los multiplexores y la conversión A/D.

El primer paso es definir las variables necesarias para desarrollar y detectar un cambio en la lectura del sensor. Estas variables son:

- `UNSIGNED16 uIRDistance`, que almacena la lectura del sensor una vez que es convertido digitalmente.
- `static UNSIGNED16 uIRDistanceOld`, guarda la lectura anterior del sensor para detectar cambios con respecto a la actual (`uIRDistance`).

El segundo paso es realizar la configuración de la señal PWM, necesaria para excitar el emisor del sensor óptico, con las características explicadas y como se detallaba en el punto 3.3.2.

El siguiente paso es realizar un barrido de lectura del anillo anticolidión, como se explico en el apartado 3.3.4. Tras realizarla se chequea si se han producido cambios con respecto a la lectura anterior. En caso afirmativo esto es indicado mediante un bit de estado, el nuevo valor es almacenado en los buffers de transmisión y la nueva posición es guardada en `uIRDistanceOld`. A continuación se muestra el fragmento de código al que se hace referencia en la explicación anterior:

```
...  
// Si se produce un cambio de posición activar interrupción  
if (uIRDistanceOld.word != uIRDistance.word)  
{  
    uIRState.bits.b1 = 1;  
    uIRState.bits.b4 = 1;  
    *(UNSIGNED16 *)uLocalTPDO2Buffer = *(UNSIGNED16 *)uLocalTPDO1Buffer =  
    *((UNSIGNED16 *)&uIRDistance.word);  
    uIRDistanceOld.word = uIRDistance.word;  
}  
...
```

Existe en el programa una variable global (`uIRState`) que nos indica el estado de los TPDO's según se explica a continuación:

- `b0`, indica que el TPDO_1 ha de ser enviado.
- `b1`, indica al TPDO_1 que la lectura del encoder a variado.
- `b2`, cuando la variable `uIRSyncSet_1 = 0` indica al TPDO_1 que la lectura esta lista para ser enviada cuando se reciba un SYNC Object.
- `b3`, indica que el TPDO_2 ha de ser enviado.
- `b4`, indica al TPDO_2 que la lectura del encoder a variado.
- `b5`, cuando la variable `uIRSyncSet_2 = 0` indica al TPDO_2 que la lectura esta lista para ser enviada cuando se reciba un SYNC Object.
- `b6` y `b7`, no se utilizan.

Una vez que se ha detectado un cambio de distancia en la lectura de alguno de los sensores, el siguiente paso es chequear el modo de sincronismo y enviar la lectura. Si estamos en modo asíncrono (`uIRSyncSet` igual a 254 ó 255) envía la lectura inmediatamente y en el modo síncrono (`uIRSyncSet` en el rango [1...253]) espera la recepción de tantos mensajes de sincronismo como indique el valor de `uIRSyncSet`.

Existe la posibilidad de que se le asigne a la variable `uIRSyncSet` el valor '0' en cuyo caso cumple la misma funcionalidad que cuando se le asigna el valor '1' pero sólo transmitirá el mensaje si se ha producido un cambio de posición antes de recibir el SYNC Object.

```

:
if (uIRState.bits.b1)

    switch (uIRSyncSet_1)
    {
        case 0:                                     // Asyclic synchronous transmit
            // Set a synchronous transmit flag
            uIRState.bits.b2 = 1;
            break;

        case 254:                                     // Asynchronous transmit
        case 255:
            // Reset the asynchronous transmit flag
            uIRState.bits.b0 = 1;
            break;
    }
:

```

El último paso es la activación de un flag para que el dato sea enviado por el NMT y el reset de los flag indicadores de envío de datos pendientes.

```

:
// If ready to send
if (mTPDOIsPutRdy(1) && uIRState.bits.b0)
{
    // Tell the stack data is loaded for transmit
    mTPDOWritten(1);

    // Reset any synchronous or asynchronous flags
    uIRState.bits.b0 = 0;
    uIRState.bits.b1 = 0;
}
:

```

unsigned int conversion (unsigned char canal)

Esta función nos permite realizar la conversión A/D, de las señales analógicas que nos llegan desde los sensores. Los pasos de la conversión están explicados en el apartado 3.3.3 de este documento.

```

unsigned int conversion (unsigned char canal)
{
    SetChanADC (canal); // selecciona el canal
    Delay10TCY x(2); // Retardo, porque después de cada conversión necesita un retardo antes de comenzar
    la siguiente
    ConvertADC(); // Comienza la conversión
    while (BusyADC ()); // Espera a terminar la conversión
    return ReadADC(); // Devuelve el resultado de la conversión
}

```

void CO_COMMSyncEvent(void)

Es la función encargada del tratamiento de sincronismos de la comunicación. En este caso se editará, para el caso del TPDO_1, de la forma que muestra el código siguiente:

```
// TPDO1
// Procesar sólo en modo síncrono
if ((uIRSyncSet_1 == 0) && (uIRState.bits.b2))
{
    // Activar flag para envío de mensaje.
    uIRState.bits.b2 = 0;
    uIRState.bits.b0 = 1;
}
else
{
    if ((uIRSyncSet_1 >= 1) && (uIRSyncSet_1 <= 240))
    {
        // Cuenta de mensajes de sincronismo para el modo síncrono.
        uIRSyncCount_1--;
        // Habilitar envío de mensaje.
        if (uIRSyncCount_1 == 0)
        {
            // Reset del contador de mensajes de sincronismo.
            uIRSyncCount_1 = uIRSyncSet_1;
            // Activar flag de transmisión.
            uIRState.bits.b0 = 1;
        }
    }
}
```

void AppLED(void)

Esta función controlara el LED con el fin de indicar el estado del Nodo. En estado STOPPED el LED permanecerá apagado, parpadeara en PRE-OPERATIONAL y permanecerá encendido mientras este en OPERATIONAL.

```
void AppLED(void)
{
    if (COMM_STATE_STOP)
    {
        LATCbits.LATC1 = 0;
    }else if (COMM_STATE_OPER)
    {
        LATCbits.LATC1 = 1;
    }else
    {
        if (ContLED <= 0)
        {
            ContLED += LED_PERIOD_MS;
            LATCbits.LATC1 = ~LATCbits.LATC1;
        }else
        {
            ContLED -= CO_TICK_PERIOD;
        }
    }
}
```


}

Donde:

- LED_PERIOD_MS: Indica la frecuencia con la que el LED parpadea.
- CO_TICK_PERIOD: Indica con qué frecuencia en milisegundos es llamada la función.

Capítulo 4

Ensayos y resultados

En este capítulo se explican los distintos ensayos que se han realizado para conseguir la implementación de un sistema anticolisión lo más cercano posible a las necesidades que se planteaban.

Se han realizado pruebas a los distintos circuitos que componen el sistema anticolisión y pruebas de comunicación por CANbus.

4.1 Pruebas y ensayos sobre el sensor infrarrojo.

Una vez seleccionado el sensor infrarrojo fue necesario realizar diferentes pruebas de medición a distintas distancia para ver si correspondían a las especificaciones técnicas que señalaba su hoja de características.

Las pruebas de medición se han realizado a diferentes distancias para comprobar la respuesta del sensor en cada una de ellas. El objeto a detectar es una tarjeta de cartón, de color blanco colocado en un calibre para que nos de información de la señal analógica que nos proporciona el sensor dependiendo de la cercanía del objeto. Cabe destacar que esta prueba se ha realizado mediante la excitación del led emisor a 300mA como se detalla la figura 65:

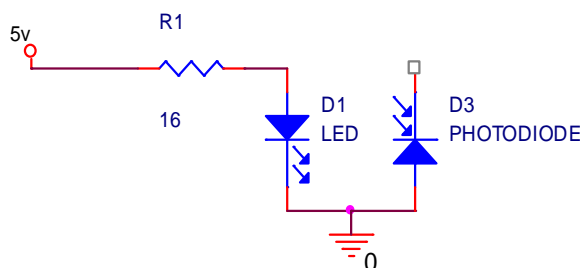


Figura 65. Esquema prueba del sensor infrarrojo

El ensayo se ha realizado en un aula oscura, debido a que el fotodiodo era sensible a la luz artificial y sin objetos que detectar existía una señal de ruido de aproximadamente 50Hz, que es el equivalente al parpadeo de la luz de un fluorescente.

Se han realizado medidas a 5mm y 60mm, los limites de detección del sensor, y la respuesta para cada una de las medidas realizadas se puede observar en las figuras 66 y 67 respectivamente.

- Distancia al objeto: 5mm

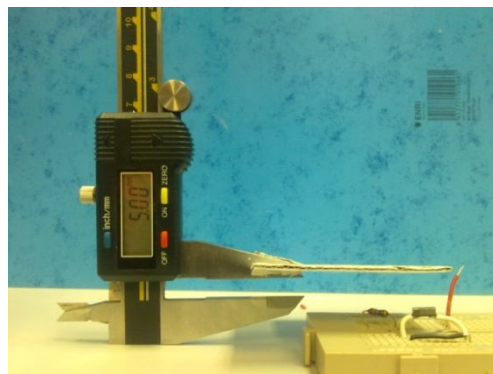
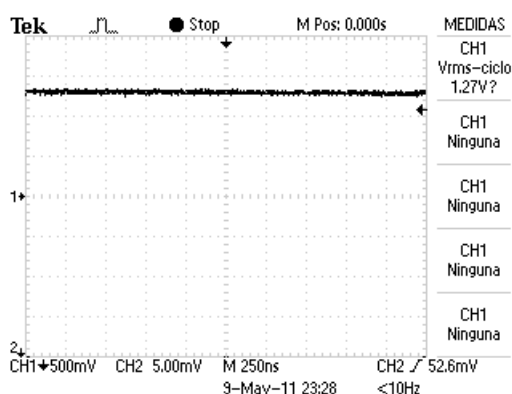


Figura 66. Respuesta del sensor con un objeto a 5mm

En la figura podemos observar que la respuesta del sensor, con el objeto a 5mm de distancia, es una señal de algo más de 1V

- Distancia al objeto: 60mm

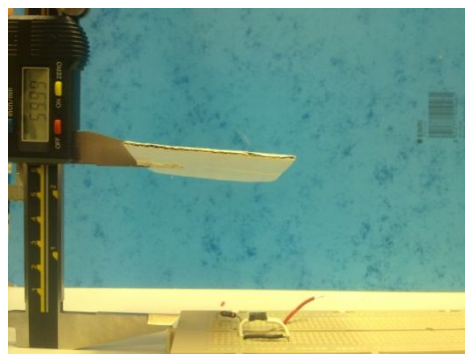
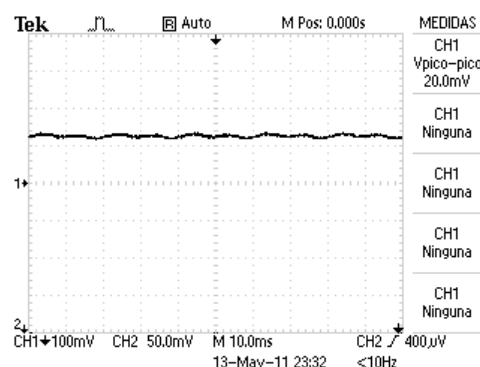


Figura 67. Respuesta del sensor con un objeto a 60mm

En la figura 68 se observa como la respuesta del sensor es una señal muy débil de aproximadamente 20mV.

4.2 Pruebas y ensayos sobre el Circuito sensor.

Las pruebas realizadas sobre el sensor infrarrojo nos indican que es necesaria la utilización de un acondicionador de señal.

El sensor infrarrojo junto con el acondicionador, mas componentes necesarios para una correcta configuración, dan lugar al circuito sensor, cuya explicación podemos encontrar en el apartado 3.2.1 y cuyo esquema podemos ver en el apartado 3.2.4 de este documento.

El ensayo es el mismo que el realizado al sensor infrarrojo, con la diferencia de que al colocar el acondicionador de señal se ha podido realizar las mediciones en un aula bajo luz artificial. En este ensayo se ha alimentado el circuito sensor a 3,3V y se ha excitado el led emisor mediante un generador de señal que ha simulado la PWM, que mas tarde se encargara el microcontrolador de generar, de 10KHz, 20% de ciclo de trabajo y 3,3V de amplitud.

Se han realizado medidas a 5mm y 117mm y la respuesta para cada una de las medidas realizadas se puede observar en las figuras 68 y 69 respectivamente.

- Distancia al objeto: 5mm

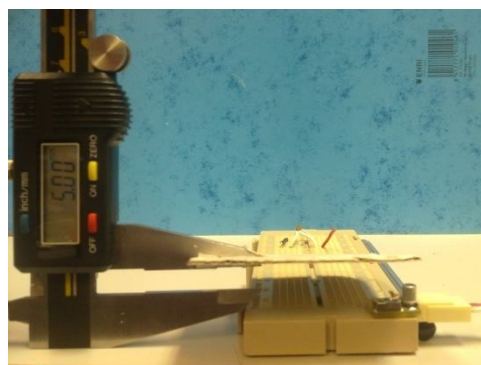
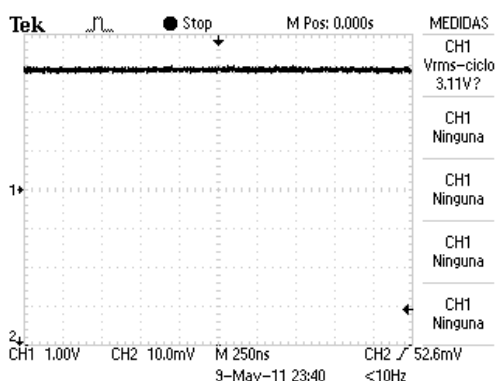


Figura 68. Respuesta del circuito sensor con un objeto a 5mm

En la figura vemos como la señal de respuesta del circuito sensor es una señal de aproximadamente 3V, una señal amplificada. El ensayo se ha podido realizar en un aula con luz artificial.

- Distancia al objeto: 117mm

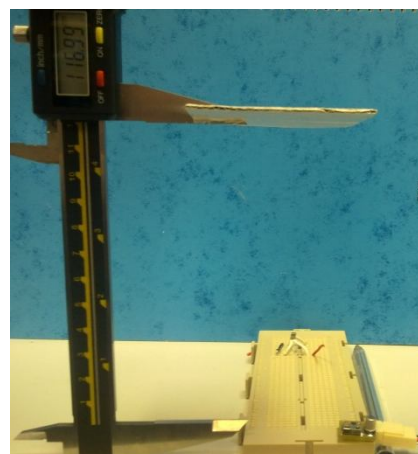
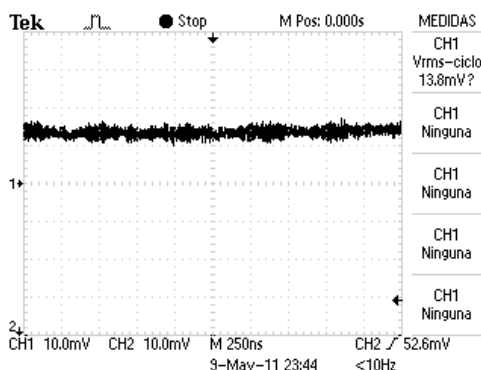


Figura 69. Respuesta del circuito sensor con un objeto a 117mm

La señal de respuesta que obtenemos a esta distancia es una señal débil pero demuestra que mediante el acondicionador de señal podemos realizar detecciones de objetos a mayor distancia.

4.3 Ensayos Circuito de Control

En este apartado vamos a ver los ensayos realizados al circuito de control, pero para ello debemos conectar un anillo sensor de prueba. Se coloca un anillo compuesto por cuatro circuitos sensores y se van a realizar las pruebas tanto de comunicación entre circuito sensor y circuito de control, como pruebas de visualización led cuando el sistema considera que nos encontramos a una distancia de peligro con el objeto detectado.

4.3.1 Generación de la señal PWM

Ya se ha explicado con anterioridad que el sensor se excita mediante una señal PWM que hasta ahora le habíamos proporcionado a través de un generador de señal, sin embargo debemos proporcionarle esta señal desde el circuito de control y para comprobar que esto es así se ha realizado un sencillo ensayo que consisten en visualizar la señal de salida por el puerto CCP en un osciloscopio. En la figura 70 se muestra una captura del osciloscopio con la señal PWM que genera nuestro sistema de control.

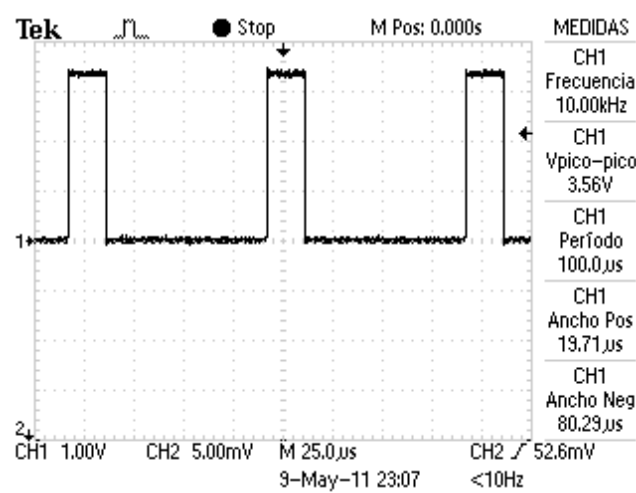


Figura 70. Señal pwm que genera el circuito de control

El resultado es una señal de 10Khz, con una amplitud de 3,3v y un ciclo de trabajo del 20%, tal y como pedían las especificaciones técnicas del acondicionador de señal utilizado para el circuito sensor.

4.3.2 Visualización led

Para el correcto funcionamiento del sistema anticolidión es primordial que exista una comunicación entre los circuitos sensores que forman el anillo y el sistema que se encarga tanto de controlarlo como de tratar los datos que recibe de ellos.

Como ya se ha explicado con anterioridad, la comunicación entre los distintos circuitos sensores y el circuito de control se realiza mediante el puerto del convertidor A/D.

Mediante el ensayo de visualización led comprobaremos que existe comunicación entre los circuitos sensores y el circuito de control y ademas comprobaremos la funcionalidad del sistema completo. Para ello se ha establecido una distancia umbral considerada distancia de peligro. Se trata de una distancia en la que si alguno de los circuitos sensores detectan algún objeto este será considerado como una amenaza para ASIBOT y este deberá detenerse para evitar la colisión. Como el sistema no está montado sobre ASIBOT se han colocado un diodo led rojo que se encenderá o apagará en función de si existe un objeto a una distancia igual o inferior a la distancia umbral.

Para este ensayo el programa ha sido adaptado, ya que se ha añadido una sentencia condicional que compara si alguna de las señales enviadas por los sensores nos indica que esta en esta zona de peligro da orden de encender o apagar el led. El fragmento de programa que se encarga de este ensayo se muestra a continuación:

```

:
for (i=0;i<4;i++)
  for (j=0;j<4;j++)
  {
    distanciar[i]=conversion(channel[j]);
    if (distanciar[i]>61)
    {
      PORTCbits.RC3 = 1;//enciende led
      uIRDistance.word=distanciar[i];
    }
    else
      PORTCbits.RC3 = 0;//led apagado
  }
:

```

Para realizar el ensayo se aproximaba al anillo sensor objetos para que este realizara la detección y encendiera el led. Un ejemplo del ensayo se muestra en la figura 71.

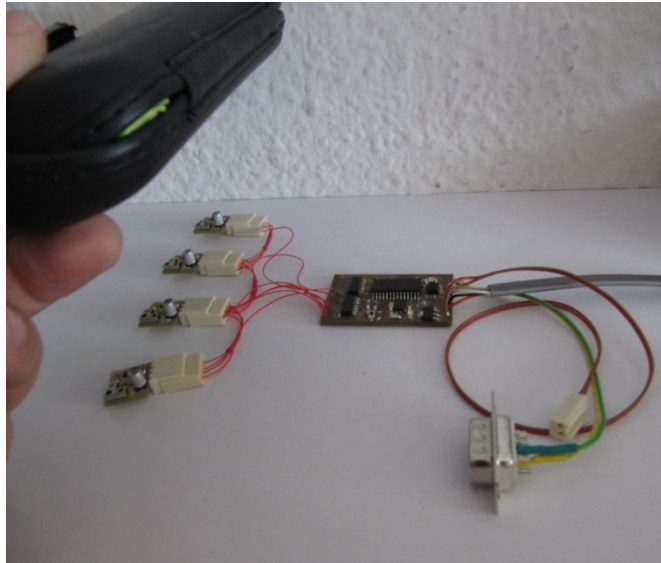


Figura 71. Detección de objetos y visualización led

Para este ensayo se aproximaron diferentes objetos para comprobar la calidad de detección. Se realizó la detección de objetos de diferentes materiales como cartón, plástico, cristal y metálicos. Otro factor a tener en cuenta a parte del material que compone el objeto es el color, por ello los objetos testeados eran de diferentes colores. De este ensayo se han obtenido resultados esperanzadores ya que en todos los casos los objetos son detectados, sin embargo hay que detectar que los objetos la distancia de detección del objeto se reduce si este es traslucido como el cristal o algunos plásticos y si el objeto es de un color oscuro como puede ser el negro y azules muy oscuros.

El resultado de este ensayo demuestra que existe comunicación entre los circuitos sensores y el circuito de control y por tanto esto no permite tomar decisiones sobre si el objeto detectado se encuentra dentro de la zona de peligro y enviar dicha distancia al ASIBOT en cuanto nos de permiso para enviar o el bus es vacío..

4.4 Pruebas en sistemas de comunicación CAN

El último ensayo realizado al sistema anticollision implementado es un ensayo de comunicación CAN con un sistema real. Para el ensayo se ha utilizado un sistema que está formado por un micro PC y una tarjeta de red CAN para comunicar el micro PC con los distintos dispositivos CAN, con un sistema de control implementado en lenguaje C bajo Linux. La figura 72 muestra el sistema de control descrito anteriormente.

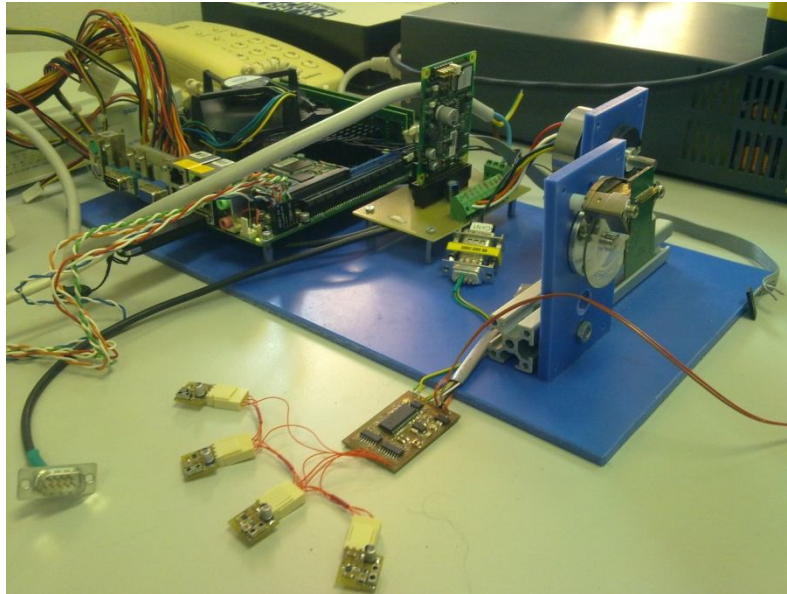


Figura 72. Sistema anticolidión conectado al sistema de control CAN

4.4.1 Configuración y resultado

Un dispositivo CAN requiere una configuración previa para poder funcionar en un sistema de control como el mostrado en la figura anterior. La configuración consiste en enviar una trama Start Remote Node que permite al NMT Maestro activar el modo Operational del NMT Esclavo, en este caso del sistema anticolidión. La trama Start Remote Node está formada por un identificador y 2 bytes de datos. El primer byte transmite la petición de activación del modo Operational y el segundo el Node ID identificativo del dispositivo que lo solicita.

Además de activar el nodo para que pueda desempeñar sus funciones, en muchos casos es necesaria la configuración de parámetros y activación de ciertas características para que el nodo pueda realizar todas las tareas requeridas por el sistema. Estas configuraciones se realizan a través de SDOs y en este caso consiste en configurar el modo de sincronismo si fuera necesario, abrir los TPDOs que van a ser utilizados y cerrarlos en el caso de que ya no sean necesarios.

El modo de sincronismo depende directamente de la variable *uIRSyncSet* y esta puede ser modificada a través del sub-index 0x02 del Objeto de Diccionario 1800h para el TPDO_1 y 1801h para el TPDO_2. El valor de esta variable indica:

- '0', modo acíclico síncrono. Si se ha producido un evento antes del SYNC Object se transmite el PDO.
- [1... 253], modo síncrono. Transmisión de PDO después de recibir tantos mensajes de sincronismo como indique el valor de *uIRSyncSet*.

- '254' y '255', modo asíncrono. Envía un TPDO cada vez que se detecta un cambio de posición.

Los parámetros de apertura de los TPDO se muestran en las tablas 25 y 26.

Tabla 26. Parámetros del TPDO1

COB-ID	COMANDO	L_INDEX	H_INDEX	SUB-INDEX	D1	D2	D3	D4
0x600+NodeID	0x23	0x00	0x18	0x01	0x80+NodeID	0x01	0x00	0x40

Tabla27. Parámetros del TPDO2

COB-ID	COMANDO	L_INDEX	H_INDEX	SUB-INDEX	D1	D2	D3	D4
0x600+NodeID	0x23	0x01	0x18	0x01	0x80+NodeID	0x02	0x00	0x40

Donde:

- COB-ID, es igual a 600h más NodeID.
- COMANDO, indica un download request con 4 bytes de datos.
- L_INDEX, es la parte baja del índice.
- H_INDEX, es la parte alta del índice.
- SUB-INDEX, es el sub-índice del Objeto.
- D1, representa la parte baja del COB-ID del TPDO.
- D2, representa la parte alta del COB-ID del TPDO.
- D3, siempre 0x00.
- D4, comando abrir TPDO.

La herramienta para cerrar un TPDO el comando en D4 que sería 0xC0.

Los datos de la distancia al objeto se transmiten a través de los TPDO con una estructura de trama 180h + NodeID + Dato(2bytes), para el TPDO1 y con una estructura 280h + NodeID + Dato(2bytes) para el TPDO2.

El resultado del ensayo es satisfactorio ya que existe comunicación entre el sistema de control y nuestro sistema anticolidión.

Capitulo 5

Conclusiones y futuros desarrollos

A lo largo de este capítulo se realiza una evaluación de los resultados obtenidos y se detallaran las conclusiones finales sobre el sistema anticolidión implementado. Se propondrán futuras ampliaciones y se realizara un análisis crítico.

5.1 Conclusiones

Para cumplir el objetivo del diseño, descrito en el apartado 1.2 de este documento, ha sido necesario seguir una serie de pasos, que podríamos llamar, objetivos intermedios que se detallan a continuación:

1. Estudio de diferentes tipos de sensores de proximidad sin contacto y estudio de otros proyectos que utilizan un sistema anticolidión o de detección de objetos.

En el apartado 2.2 de este documento se puede observar un exhaustivo estudio de diferentes sensores de proximidad sin contacto lo que demuestra que la elección del sensor no ha sido una decisión tomada a la ligera. Se han tenido en cuenta diferentes factores como la distancia de detección, zonas muertas y requisitos eléctricos que cumplieran las condiciones para poder incorporarlos a ASIBOT.

El diseño del sistema anticolidión ha sido posible gracias al estudio de otros proyectos que incorporan sistemas anticolidión. De los proyectos estudiados se ha llegado a la conclusión de que un sistema en anillo era lo más eficiente, además en este caso la solución de un anillo enseguida encajo con la estructura física de ASIBOT, ya se podía crear un sistema con forma de pulsera al tratarse de un brazo robot.

2. Selección del tipo de sensor más adecuado y su posterior estudio e interpretación de las señales que nos proporciona.

Tras el estudio de los diferentes tipos de sensores se lleo a la conclusión de que un sensor infrarrojo de reflexión directa era el más adecuado para nuestro sistema. El sensor seleccionado es un sensor que permite la detección a distancias cortas, además es de un tamaño muy reducido y de un haz de detección de una apertura que permite crear un anillo alrededor de la pinza de ASIBOT con un menor número de sensores y no pone en peligro la integridad del robot.

3. Implementación de un circuito sensor que cumpla las características necesarias de detección.

El sensor por si solo proporciona una señal analógica en función de la distancia a la que se encuentra el objeto. Sin embargo es sensible a la luz artificial y a la luz solar, lo que provoca que el sistema sea de dudosa fiabilidad tanto en espacios cerrados como en espacios al aire libre. Para solucionarlo se utiliza un acondicionador de señal que realiza la función de filtrado de luz artificial y solar, además también realiza una amplificación de la señal recibida por el sensor, lo que aumenta la distancia de detección del sensor.

4. Elección de un microcontrolador adecuado para las necesidades de nuestro sistema.

Para el correcto funcionamiento del sistema anticolidión es necesario establecer un control sobre los circuitos sensores. Para ello se ha seleccionado el microcontrolador 18F2580 ya que reúne los requisitos necesarios. Tiene modulo convertidor A/D con una resolución de 10 bits por canal, modulo de comunicación CAN y puerto CCP para generación de señales PWM.

5. Implementar un circuito de control que permita el control y la comunicación con los circuitos sensores, y además comunicación con ASIBOT mediante protocolo CANopen.

El circuito de control estará formado por el microcontrolador y los componentes necesarios para el control del sistema. El microcontrolador seleccionado tiene 8 canales de conversión A/D, sin embargo el anillo está formado por 16 circuitos de detección de objetos, lo que hace necesario la colocación de multiplexores para la gestión de la entrada de las señales.

En el circuito de control hay un convertido de tensión para hacer llegar alimentación a los circuitos sensores.

Tienen un cristal de 20 MHz utilizado para la generación de la PWM que excitan a los diodos de luz infrarroja de los sensores.

Por último también se ha colocado un transceiver para hacer posible la emisión y recepción de datos entre el sistema anticolidión y ASIBOT.

6. Adquirir los conocimientos necesarios sobre el entorno de desarrollo MPLAB de Microchip®, y su herramienta compiladora para microcontroladores de la familia PIC18.

Par llevar el desarrollo software del sistema anticolidión fue necesario el estudio de una guía de utilización de MPLAB así como guías de programación en C18.

7. Desarrollo hardware de un sistema anticolidión de prueba en una placa de puntos.

Para la verificación de los sistemas planteado se desarrollaron placas de puntos tanto de los circuitos sensores como del circuito de control

8. Desarrollo software, en el que se incluye configuración del microcontrolador, generación de señal PWM, comunicación a través del convertidor A/D, conversión de datos, comparación de resultados con distancia de peligro, señal luminosa de peligro y envío de distancia por CANbus.

Para conseguir el control por parte del microcontrolador es necesario desarrollar un programa y guardárselo en memoria para que lo ejecute cíclicamente. Con este programa el microcontrolador realiza un barrido de detección alternando la excitación de unos sensores con otros y realizando el control de los multiplexores para gestionar la entrada de las señales por los puertos del convertidor A/D, para su posterior conversión y comparación con la zona de peligro definida para que en caso de entrar en ella emitir un mensaje de peligro y una señal luminosa.

El programa genera una señal PWM que como ya se ha comentado con anterioridad excita los sensores infrarrojos.

El programa también genera la trama que más tarde será enviada al ASIBOT mediante el transceiver en caso de peligro de colisión.

9. Comprobación del correcto funcionamiento del sistema de pruebas mediante los apartados descritos en el capítulo 4.

Es necesario realizar una validación del sistema anticolidión más exhaustiva en un entorno real con ASIBOT

10. Diseño e implementación de un sistema definitivo con componentes en SMD, que incluye el diseño de las placas en Orcad.

Con todo lo trabajado sobre este proyecto y como muestran los ensayos realizados podemos llegar a la conclusión de que el sistema cumple los objetivos propuestos. Es capaz de detectar objetos a una distancia de seguridad, almacenar dicha distancia e informar mediante CANbus a ASIBOT o cualquier sistema de control CAN.

5.2 Futuras ampliaciones

En este apartado se plantearan mejora y ampliaciones, que permitan al sistema anticolidión ser un sistema más completo y funcional.

A nivel de planteamiento quizás convenga replantearse el uso de un solo tipo de sensor. Se podría plantear un sistema que intercalara sensores infrarrojos con el otro tipo de sensor candidato a formar parte de un sistema anticolidión, sensores ultrasónicos.

Por espacio no podría utilizarse sensores ultrasónicos cuyo emisor y receptor estén en cuerpos separados y los que se componen de emisor y receptor en un solo cuerpo han sido descartados debidos a la gran zona muerta que se genera debido al funcionamiento de este tipo de sensores. Realizan la emisión de una onda a una frecuencia imperceptible para el oído humano, pasa de estado emisor a estado receptor y recibe el eco de la onda emitida.

Sin embargo esta zona muerta podría solucionarse mediante los sensores infrarrojos que componen el anillo anticolidión implementado en este proyecto. Eliminando dicha zona muerta y utilizando sensores ultrasónicos tendríamos un mayor control sobre el entorno detectando objetos a distancia mucho más largas que no habría por que tomar como peligro. Esto permitiría a ASIBOT realizar variaciones de velocidad para aproximaciones a objetos. Este cambio de diseño supondría un aumento en el consumo del dispositivo, sin embargo es posible realizar un control sobre los barridos de detección de manera que definiendo el alcance y zona de detección de cada uno de los tipos de sensores estos estuvieran coordinados para que unos funcionen hasta que comience la zona de detección de los otros, es decir, los sensores infrarrojo podrían funcionar desde los 5 mm hasta los 100 mm, debido a que el circuito sensor implementado en este proyecto lo hace posible, y los sensores ultrasónicos entrarían en funcionamiento aproximadamente a los 90mm hasta su límite de alcance, pasando los sensores infrarrojos a modo de stanby , lo cual es posible mediante el pin 3 del acondicionador de señal del circuito sensor. Alternando el funcionamiento reduciríamos el consumo del nuevo dispositivo.

Para conseguir una mayor funcionalidad al sistema desarrollado, y que fuera posible su utilización en otros robots, podríamos dotarlo de una batería, ya que su bajo consumo lo hace totalmente posible. Con una batería podríamos incorporar este sistema a cualquier robot que permita conexión mediante CANbus. Esta mejora permitiría convertir un sistema anticolidión diseñado para un robot en concreto en un sistema anticolidión útil para cualquier robot con comunicación CANbus.

Otra futura mejora o ampliación a nivel hardware seria el diseño del sistema en un circuito impreso flexible. Se ha establecido un primer contacto con empresas que se dedican a la fabricación de este tipo de placas y seria una solución bastante interesante para un futuro cuando se obtenga un diseño con una validación mas exhaustiva en un entorno real con ASIBOT. Los componentes deberán estar soldados en partes de placa rígida pero tanto las conexiones para llevar tanto las señales de alimentación, masa y la señal PWM entre ellas, como las señales de respuesta de cada uno de los circuitos sensores con el circuito de control podrían realizarse en circuito impreso flexible para que se adapte adecuadamente a cualquier posición necesaria en ASIBOT u otros lugares.

A nivel de software sería interesante una comunicación en la que ASIBOT podría enviar por CANbus la velocidad de su extremo para que nuestro sistema obtenga esa información del bus y mediante un algoritmo calcule cuál sería la distancia de peligro adecuada para detener a tiempo el robot y evitar colisiones. Esta distancia de peligro en nuestro diseño debe ser introducida por el usuario realizando modificaciones en el código y realizando la reprogramación del microcontrolador.

5.3 Análisis crítico

Se ha diseñado un sistema compacto, económico, ligero y de bajo consumo, que incorpora un módulo de conexión CANopen y permite la detección de objetos comprendidos entre los 5mm y los 150mm. El sistema, gracias a su diseño, permite ser integrado en cualquier robot con conexión CAN y una señal de alimentación de 5V. Además incorpora un sistema de programación llamado ICSP (InCircuit Serial Programming) que permite programación y depuración de código sin necesidad de ser desmontado del sistema robótico que lo porte en ese instante.

Se ha implementado un sistema muy completo y que cumple con los objetivos planteados, pero que acepta mejoras e implementaciones que podrían generar nuevas investigaciones como las que se plantean en el apartado 5.2.

Referencias

- [1] Martínez Alberto, David. "Análisis cinemático y dinámico del robot PASIBOT"; PFC Universidad Carlos III de Madrid, 2008
- [2] A. Jardón, R. Correal, S. Martínez, A. Giménez, C. Balaguer. "Robots personales y asistenciales"; RoboticsLab; Universidad Carlos III de Madrid, 2006
- [3] F. Ebel, S. Nestel. "Sensores para la técnica de procesos y manipulación", 1993
- [4] J.Ureña, M.Mazo, J.Garcia, J.Lozano, P.Revenga, J.C.Garcia, F.J.Rodriguez, E.Santiso. "Sensores de ultrasonidos e infrarrojos usados en la detección de obstáculos e irregularidades del suelo para aplicaciones de robots móviles". Departamento de Electrónica. Universidad de Alcalá, 1995.
- [5] A. Herrera Velásquez; R. Jaramillo Morales. "Diseño y construcción de un dispositivo para alerta de obstáculos". Escuela de Ingeniería de Antioquia–Universidad CES, Medellín, Colombia, 2007
- [6] Kaijen Hsiao, Paul Nangeroni, Manfred Huber, Ashutosh Saxena, Andrew Y. "Reactive Grasping using Optical Proximity Sensors", Robotics and Automation, 2009. ICRA '09. IEEE International Conference.
- [7] Fortes Monteiro, J.; Diseño e implementación encoder absoluto CANopen.; PFC Universidad Carlos III de Madrid, 2009.
- [8] Robert Bosch GmbH. "BOSCH Controller Area Network protocol specification". Revision 2.0, Stuttgart, 1991.
- [9] H. Boterenbrood. NIKHEF, Amsterdam. "CANopen: high level protocol for CAN-bus", March20, 2000. [Consulta: Agosto 2010].
<http://www.nikhef.nl/pub/departments/ct/po/doc/CANopen30.pdf>
- [10] J. Alvarez Paramio. " Diseño e implementación de una arquitectura para el control del robot humanoide RH-2".PFC Universidad Carlos III de Madrid, 2011.
- [11] Ross M. Fosler,"A CANopen Stack for PIC18 ECANTM Microcontrollers". Microchip Technology Incorporated, 2004
- [12] "MPLAB® C18 C Compiler User's Guide", Microchip Technology Inc., 2005.
- [13] "CANopen: a CAL based communication profile for industrial systems", Draft Standard DS-301. Revision 4.02, CAN in Automation Group, Febrero 2002.

Anexos

Anexo1. Listado de componentes

COMPONENTE	UNIDADES	DESCRIPCION
CIRCUITO SENSOR		
HDSL-9100	16	SENSOR INFRARROJO
APDS-9700	16	ACONDICIONADOR DE SEÑAL
RESISTENCIA 10R	16	RESISTENCIA
CONDENSADOR 100nF	16	CONDENSADOR
CONDENSADOR 6,8uF	16	CONDENSADOR
POTENCIOMETRO 500K	16	RESISTENCIA VARIABLE
CONDENSADOR 3,3nF	16	CONDENSADOR
CIRCUITO DE CONTROL		
PIC18F2580	1	MICROCONTROLADOR
MC14551B	2	MULTIPLEXOR
MCP2551	1	TRANSCEIVER
MCP1825	1	REGULADOR DE TENSION
LED	2	LED INDICADOR ROJO
CRISTAL 20MHz	1	CRISTAL 20MHz
CONDENSADOR 22Pf	2	CONDENSADOR
RESISTENCIA 150R	2	RESISTENCIA
RESISTENCIA 100R	1	RESISTENCIA
RESISTENCIA 10KR	1	RESISTENCIA
RESISTENCIA 1KR	1	RESISTENCIA
RESISTENCIA 2KR2	1	RESISTENCIA
BU3-1000P	1	SWITCH

Anexo 2. Presupuesto.

CIRCUITO SENSOR		
COMPONENTE	UND	PVP (€)
HDSL-9100	1	2,15
APDS-9700	1	0,96
RESISTENCIA 10R	1	0,03
RVAR 500KR	1	2,12
CONDENSADOR 100nF	1	0,07
CONDENSADOR 3,3nF	1	0,04
CONDENSADOR 6,8 uF	1	0,17
TOTAL		5,54
CIRCUITO DE CONTROL		
PIC18F2580	1	6,37
MC14551B	2	2,01
MCP2551	1	1,08
MCP1825	1	1,33
LED	2	0,10
CRISTAL 20MHz	1	1,15
CONDENSADOR 22pF	2	0,06
CONDENSADOR 100nf	1	0,07
RESISTENCIA 150R	2	0,38
RESISTENCIA 100R	1	0,02
RESISTENCIA 10KR	1	0,02
RESISTENCIA 1KR	1	0,13
RESISTENCIA 2KR2	1	0,13
BU3-1000P	1	1,00
TOTAL		16,4
SISTEMA ANTICOLISION		
CIRCUITO SENSOR	16	5,54
CIRCUITO DE CONTROL	1	16,4
TOTAL		105,04
FABRICACION PCB FLEXIBLE		
CONFIG. DISEÑO CTO SENSOR	1	1000
CONFIG DISEÑO CTO CONTROL	1	200
PCB CIRCUITO DE SENSOR	16	100
PCB CIRCUITO DE CONTROL	5	80
TOTAL		3200
MANO DE OBRA		
	h	PVP (€)
DISEÑO CONCEPTUAL	120	30,63
DISEÑO DETALLE CTO SENSOR	75	30,63
DISEÑO DETALLE CTO CONTROL	80	30,63
PROGRAMACION PIC	250	30,63
PUESTA EN MARCHA	5	30,63
PRUEBAS FUNCIONALES	20	30,63
TOTAL		19900

Anexo 3. Sensor infrarrojo



Agilent HSDL-9100 Miniature Surface-Mount Proximity Sensor Data Sheet



Description

The HSDL-9100 is an analog-output reflective sensor with an integrated high efficiency infrared emitter and photodiode housed in a small form factor SMD package. The optical proximity sensor is housed in a specially designed metal-shield to ensure excellent optical isolation resulting in low optical cross-talk.

HSDL-9100 is a class of its own with its small form SMD package and at a detection range from near zero to 60mm. It is specifically optimized for size, performance and ease of design in mobile constrained applications such as mobile phones and notebooks.

HSDL-9100 has extremely low dark current and high signal to noise ratio (SNR) where high SNR is achieved with a pair of highly efficient infrared emitter and highly sensitive detector.

Application Support Information

The Application Engineering Group is available to assist you with the application design associated with HSDL-9100 Proximity Sensor. You can contact them through your local sales representatives for additional details.

Order Information

Part Number	(HSDL-9100-021)
Packaging Type	(Tape and Reel)
Package	(PCB Substrate, moulded package)
Quantity	(2500 units)

Features

- Excellent optical isolation resulting in near zero optical cross-talk
- High efficiency emitter and high sensitivity photodiode for high signal-to-noise ratio
- Low cost & lead-free miniature surface-mount package
Height – 2.70 mm
Width – 2.75 mm
Length – 7.10 mm
- Detect objects from near zero to 60mm
- Low dark current
- Guaranteed Temperature Performance
-40°C to 85°C

Applications

- Mobile phones
- Notebooks
- Industrial Control
- Printers, Photocopiers and Facsimile machines
- Home Appliances
- Vending Machines

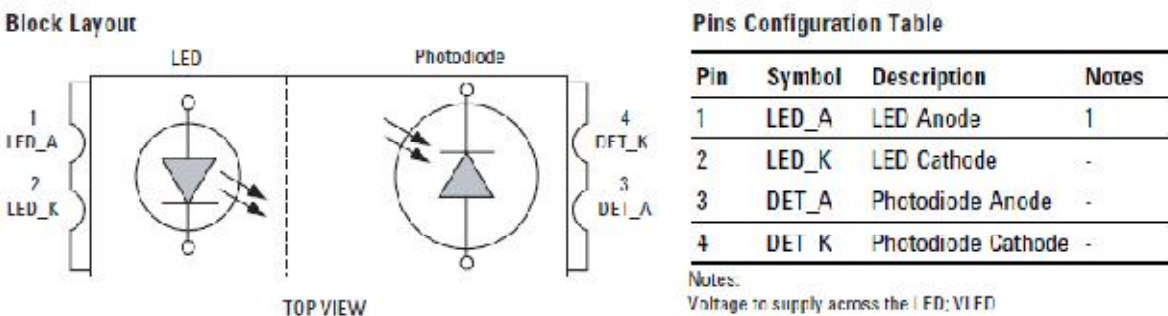


Figure 1. Block Layout of HSDL-9100

Absolute Maximum Ratings (Ta=25°C)

Parameter	Symbol	Ratings		Units
		Min.	Max	
Emitter				
Continuous Forward Current	I _{DC}		50	mA
Coupled				
Total Power Dissipation (refer to Figure 1)	P _{TOT}	-	165	mW
Operating Temperature	T _{OP}	-40	+85	°C
Storage Temperature	T _{STG}	-40	+100	°C
Reflow Soldering Temperature	T _{SOL}		260	°C

Electrical-Optical Characteristics (Ta=25°C)

Parameter	Symbol	Test Condition	Ratings			Units
			Min	Typ	Max	
Emitter						
Forward Voltage	V_f	$I_f = 100\text{mA}$		1.50	1.65	V
Reverse Voltage	V_R	$I_R = 10\mu\text{A}$	5	—	—	V
Peak Wavelength	λ_p	$I_f = 20\text{mA}$	—	940	—	nm
Spectrum Width of Half Value	$\Delta\lambda$	$I_f = 20\text{mA}$	—	50	—	nm
Detector						
Dark Current	I_{dark}	$V_R = 10\text{V}$, $L^{**} = 0$	—	2	10	nA
Forward Voltage	V_f	$I_f = 10\text{mA}$, $L = 0$	0.5		1.3	V
Reverse Breakdown Voltage	V_{BR}	$I_R = 100\mu\text{A}$, $L = 0$	—	—	35	V
Coupled						
Output Current	I_O	Refer to Fig 2	—	0.1	—	mA
Peak Output Distance	D_0	Refer Note 1	—	5	—	mm
Operating Cross Talk Current	I_{CT}	Refer to Fig 3	—	—	200	nA
Rise Time (LED)	T_{RL}	$R_L = 50\Omega$		50		ns
Fall Time (LED)	T_{FL}	$R_L = 50\Omega$	—	50	—	ns
Rise Time (Photodiode)	T_{RD}	$R_L = 5.1\text{K}\Omega$	—	6	—	μs
Fall Time (Photodiode)	T_{FD}	$R_L = 5.1\text{K}\Omega$	—	6	—	μs

** L = 0 (zero light condition)

Note:

1. I_{Lcd} = 300mA Pulse, 5% Duty Cycle
(Kodak 18% Reflectance Gray Card)

Anexo 4. Acondicionador de señal

APDS-9700

Signal Conditioning IC for Optical Proximity Sensors



Data Sheet

Description

APDS-9700 is a signal conditioning IC that enhances the performance and robustness of the optical sensors used for proximity or object detection.

APDS-9700 is a single chip solution that consists of a LED driver circuit, sunlight cancellation and built-in LED stuck high protection circuit integrated into a single chip. APDS-9700 has artificial light immunity and is also operational under the sun. Design flexibility is optimized as APDS-9700 can be paired up with an integrated proximity sensor or discrete pair solution.

APDS-9700 can be disabled to maximize power savings and battery life in applications such as portable or battery-operated devices. The LED current of the optical proximity sensors can be configured to different levels using a limiting resistor at the LEDA pin. APDS-9700 also provides user flexibility to control the pulse width with suitable burst rate, duty cycle and frequency that can reduce power consumption. These low power consumption features makes it also ideal for low power mobile and handheld devices.

APDS-9700 is capable of operating at voltage supply ranging from 2.4 V to 3.6 V. APDS-9700 has two separate output pins for analog and digital outputs. This provides flexibility to use either the analog or digital output (or both) depending on the requirements of the application.

The device is packaged in 8-pin QFN package measuring 0.55mm(H) x 2mm(W) x 2mm(L).

Ordering Information

Part Number	Package	Shipping Option
APDS-9700-020	Tape & Reel	2500

Application Support Information

The Application Engineering Group is available to assist you with the application design associated with APDS-9700 module. You can contact them through your local sales representatives for additional details.

Features

- Low power consumption
 - LED pulse width control
 - Low shut down current
 - External LED drive-current control
- Complete shutdown mode
- Supply voltage : 2.4 V to 3.6 V
- Operational in sunlight conditions up to 100klux(with HSDL-9100)
- Artificial light immunity
- Analog & Digital output available
 - Built in hysteresis comparator for digital output
- LED stuck High protection
- Wide bandwidth Trans-impedance amplifier
- External capacitor and resistor for integration and gain controls
- Flexibility to enhance detection distance up to 200mm with HSDL-9100 or further with external discretes pair
- Small 2mm x 2mm QFN 8-pin package
- Design flexibility to pair with Avago Proximity Sensors or discretes pair solution
- Lead-free & ROHS Compliant

Applications

- PDA and mobile phones
- Portable and Handheld devices
- Personal Computers/Notebooks
- Amusement/Games/Vending Machines
- Industrial Automation
- Contactless Switches
- Sanitary Automation

Absolute Maximum Ratings (Ta=25°C)

Parameter	Symbol	Min.	Max.	Units	Conditions
Supply Voltage	V _{CC}	0	4.5	V	
Input Logic Voltage	V _I	0	4.5	V	
Reflow Soldering Temperature			260	°C	

Recommended Operating Conditions

Parameter	Symbol	Min.	Max.	Units	Conditions
Operating Temperature	T _A	-40	105	°C	
Storage Temperature	T _S	-40	125	°C	
Supply Voltage	V _{CC}	2.4	3.6	V	

Electrical & Optical Specifications (Ta=25°C)

Parameters	Symbol	Minimum	Typical	Maximum	Units	Conditions
Input						
Logic High Voltage, LEDON	V _{IH}	1.6		V _{CC}	V	
Logic High Voltage, ENB	V _{IH}	1.4		V _{CC}	V	For V _{CC} = 2.4V
		1.5		V _{CC}	V	For 2.4V < V _{CC} ≤ 3V
		1.7		V _{CC}	V	For 3V < V _{CC} ≤ 3.6V
Logic Low Voltage, LEDON	V _{IL}	0		0.3	V	
Logic Low Voltage, ENB	V _{IL}	0		0.3	V	
Logic High Input Current, LEDON	I _{IH}		0.1	1	μA	V _I ≥ V _{IH}
Logic High Input Current, ENB	I _{IH}		0.1	1	μA	V _I ≥ V _{IH}
Logic Low Input Current, LEDON	I _{IL}		0.1	1	μA	V _I ≤ V _{IL}
Logic Low Input Current, ENB	I _{IL}		0.1	1	μA	V _I ≤ V _{IL}
Shutdown Current	I _{SD}		0.3	1	μA	V _{CC} =3V, ENB=3V
Idle Current	I _{CC}		500	650	μA	V _{CC} =3V, ENB=0V
Output						
Digital Output	V _{OL}	0		0.3	V	I _{DOUT(LOW)} = 2mA, V _{CC} = 3V
Rise Time(DOUT)	T _R		1		μs	V _{CC} = 3V, R ₂ = 10kΩ, Frequency = 10kHz
Fall Time(DOUT)	T _F		1		μs	V _{CC} = 3V, R ₂ = 10kΩ, Frequency = 10kHz
Transmitter						
Rise Time (LEDA)	T _R		40		ns	V _{CC} = 3V, I _{LED} = 120mA, Freq = 10kHz
Fall Time (LEDA)	T _F		40		ns	V _{CC} = 3V, I _{LED} = 120mA, Freq = 10kHz
Max I _{LED} Pulse Width	Max-PW		120		μs	V _{CC} =3V, ENB=0V
I _{LED} Pulse Current	I _{LED}		120	300	mA	V _{CC} =3V, R ₁ = 10Ω

Anexo 5. Microcontrolador



MICROCHIP PIC18F2480/2580/4480/4580

28/40/44-Pin Enhanced Flash Microcontrollers with ECAN™ Technology, 10-Bit A/D and nanoWatt Technology

Power Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μ A typical
- Sleep mode current down to 0.1 μ A typical
- Timer1 Oscillator: 1.1 μ A, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, up to 40 MHz
- 4X Phase Lock Loop (PLL) – available for crystal and internal oscillators)
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 MHz
- Internal oscillator block:
 - 8 user selectable frequencies, from 31 kHz to 8 MHz
 - Provides a complete range of clock speeds, from 31 kHz to 32 MHz when used with PLL
 - User tunable to compensate for frequency drift
- Secondary oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor
 - Allows for safe shutdown if peripheral clock stops

Special Microcontroller Features:

- C compiler optimized architecture with optional extended instruction set
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Wide operating voltage range: 2.0V to 5.5V

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupts
- One Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI™ (all 4 modes) and I²C™ Master and Slave modes
- Enhanced Addressable USART module
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-Wake-up on Start bit
 - Auto-Baud detect
- 10-bit, up to 11-channel Analog-to-Digital Converter module (A/D), up to 100 Ksps
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual analog comparators with input multiplexing

ECAN Module Features:

- Message bit rates up to 1 Mbps
- Conforms to CAN 2.0B ACTIVE Specification
- Fully backward compatible with PIC18XXX8 CAN modules
- Three modes of operation:
 - Legacy, Enhanced Legacy, FIFO
- Three dedicated transmit buffers with prioritization
- Two dedicated receive buffers
- Six programmable receive/transmit buffers
- Three full 29-bit acceptance masks
- 16 full 29-bit acceptance filters w/ dynamic association
- DeviceNet™ data byte filter support
- Automatic remote frame handling
- Advanced error management features

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		USART	Comp.	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI™	Master I ² C™			
PIC18F2480	16K	8192	768	256	25	8	1/0	Y	Y	1	0	1/3
PIC18F2580	32K	16384	1536	256	36	8	1/0	Y	Y	1	0	1/3
PIC18F4480	16K	8192	768	256	25	11	1/1	Y	Y	1	2	1/3
PIC18F4580	32K	16384	1536	256	36	11	1/1	Y	Y	1	2	1/3

Anexo 6. Multiplexor.

MC14551B

Quad 2-Channel Analog Multiplexer/Demultiplexer

The MC14551B is a digitally-controlled analog switch. This device implements a 4PDT solid state switch with low ON impedance and very low OFF Leakage current. Control of analog signals up to the complete supply voltage range can be achieved.

- Triple Diode Protection on All Control Inputs
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Analog Voltage Range ($V_{DD} - V_{EE}$) = 3.0 to 18 V
Note: V_{EE} must be $\leq V_{SS}$
- Linearized Transfer Characteristics
- Low Noise — 12 nV $\sqrt{\text{Cycle}}$, $f \geq 1.0$ kHz typical
- For Low R_{ON} , Use The HC4051, HC4052, or HC4053 High-Speed CMOS Devices
- Switch Function is Break Before Make

MAXIMUM RATINGS (2.)

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage Range (Referenced to V_{EE} , $V_{SS} \geq V_{EE}$)	- 0.5 to + 18.0	V
V_{in} , V_{out}	Input or Output Voltage (DC or Transient) (Referenced to V_{SS} for Control Input & V_{EE} for Switch I/O)	- 0.5 to $V_{DD} + 0.5$	V
I_{in}	Input Current (DC or Transient), per Control Pin	± 10	mA
I_{sw}	Switch Through Current	± 25	mA
P_D	Power Dissipation, per Package (3.)	500	mW
T_A	Ambient Temperature Range	- 55 to + 125	°C
T_{stg}	Storage Temperature Range	- 65 to + 150	°C
T_L	Lead Temperature (8-Second Soldering)	260	°C

2. Maximum Ratings are those values beyond which damage to the device may occur.

3. Temperature Derating:

Plastic "P" and D/DW" Packages: - 7.0 mW/°C From 65°C To 125°C

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{in} and V_{out} should be constrained to the range $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$ for control inputs and $V_{EE} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$ for Switch I/O.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} , V_{EE} or V_{DD}). Unused outputs must be left open.



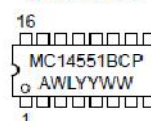
ON Semiconductor

<http://onsemi.com>

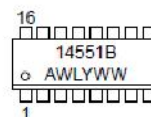
MARKING DIAGRAMS



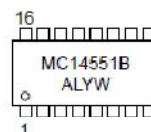
PDIP-16
P SUFFIX
CASE 648



SOIC-16
D SUFFIX
CASE 751B



SOEIAJ-16
F SUFFIX
CASE 966



A = Assembly Location
WL, L = Wafer Lot
YY, Y = Year
WW, W = Work Week

ORDERING INFORMATION

Device	Package	Shipping
MC14551BCP	PDIP-16	2000/Box
MC14551BD	SOIC-16	48/Rail
MC14551BDR2	SOIC-16	2500/Tape & Reel
MC14551BF	SOEIAJ-16	See Note 1.

1. For ordering information on the EIAJ version of the SOIC packages, please contact your local ON Semiconductor representative.

Anexo 7. Transceiver



MCP2551

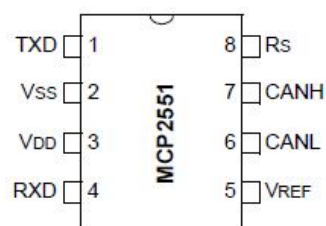
High-Speed CAN Transceiver

Features

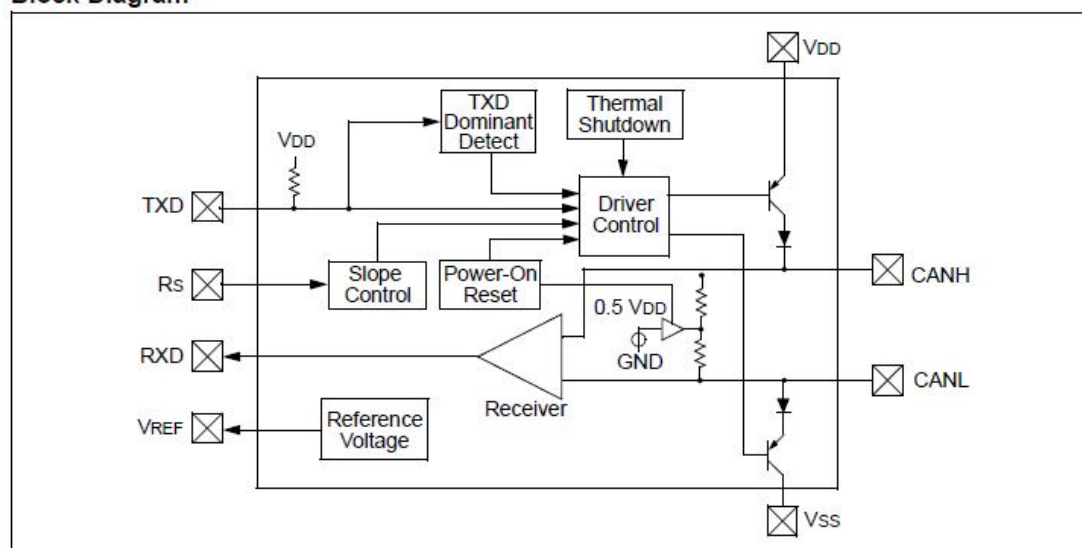
- Supports 1 Mb/s operation
- Implements ISO-11898 standard physical layer requirements
- Suitable for 12V and 24V systems
- Externally-controlled slope for reduced RFI emissions
- Detection of ground fault (permanent dominant) on TXD input
- Power-on reset and voltage brown-out protection
- An unpowered node or brown-out event will not disturb the CAN bus
- Low current standby operation
- Protection against damage due to short-circuit conditions (positive or negative battery voltage)
- Protection against high-voltage transients
- Automatic thermal shutdown protection
- Up to 112 nodes can be connected
- High noise immunity due to differential bus implementation
- Temperature ranges:
 - Industrial (I): -40°C to +85°C
 - Extended (E): -40°C to +125°C

Package Types


PDIP/SOIC



Block Diagram



Anexo 8.Regulador de tension.



MICROCHIP MCP1825/MCP1825S

500 mA, Low Voltage, Low Quiescent Current LDO Regulator

Features

- 500 mA Output Current Capability
- Input Operating Voltage Range: 2.1V to 6.0V
- Adjustable Output Voltage Range: 0.8V to 5.0V (MCP1825 only)
- Standard Fixed Output Voltages:
 - 0.8V, 1.2V, 1.8V, 2.5V, 3.0V, 3.3V, 5.0V
- Other Fixed Output Voltage Options Available Upon Request
- Low Dropout Voltage: 210 mV Typical at 500 mA
- Typical Output Voltage Tolerance: 0.5%
- Stable with 1.0 μ F Ceramic Output Capacitor
- Fast response to Load Transients
- Low Supply Current: 120 μ A (typical)
- Low Shutdown Supply Current: 0.1 μ A (typical) (MCP1825 only)
- Fixed Delay on Power Good Output (MCP1825 only)
- Short Circuit Current Limiting and Overtemperature Protection
- TO-263-5 (DDPAK-5), TO-220-5, SOT-223-5 Package Options (MCP1825).
- TO-263-3 (DDPAK-3), TO-220-3, SOT-223-3 Package Options (MCP1825S).

Applications

- High-Speed Driver Chipset Power
- Networking Backplane Cards
- Notebook Computers
- Network Interface Cards
- Palmtop Computers
- 2.5V to 1.XV Regulators

Description

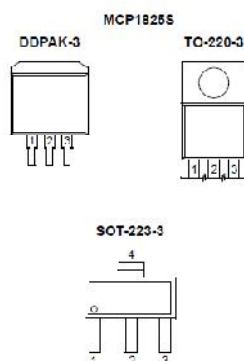
The MCP1825/MCP1825S is a 500 mA Low Dropout (LDO) linear regulator that provides high current and low output voltages. The MCP1825 comes in a fixed or adjustable output voltage version, with an output voltage range of 0.8V to 5.0V. The 500 mA output current capability, combined with the low output voltage capability, make the MCP1825 a good choice for new sub-1.8V output voltage LDO applications that have high current demands. The MCP1825S is a 3-pin fixed voltage version.

The MCP1825/MCP1825S is stable using ceramic output capacitors that inherently provide lower output noise and reduce the size and cost of the entire regulator solution. Only 1 μ F of output capacitance is needed to stabilize the LDO.

Using CMOS construction, the quiescent current consumed by the MCP1825/MCP1825S is typically less than 120 μ A over the entire input voltage range, making it attractive for portable computing applications that demand high output current. The MCP1825 versions have a Shutdown (SHDN) pin. When shut down, the quiescent current is reduced to less than 0.1 μ A.

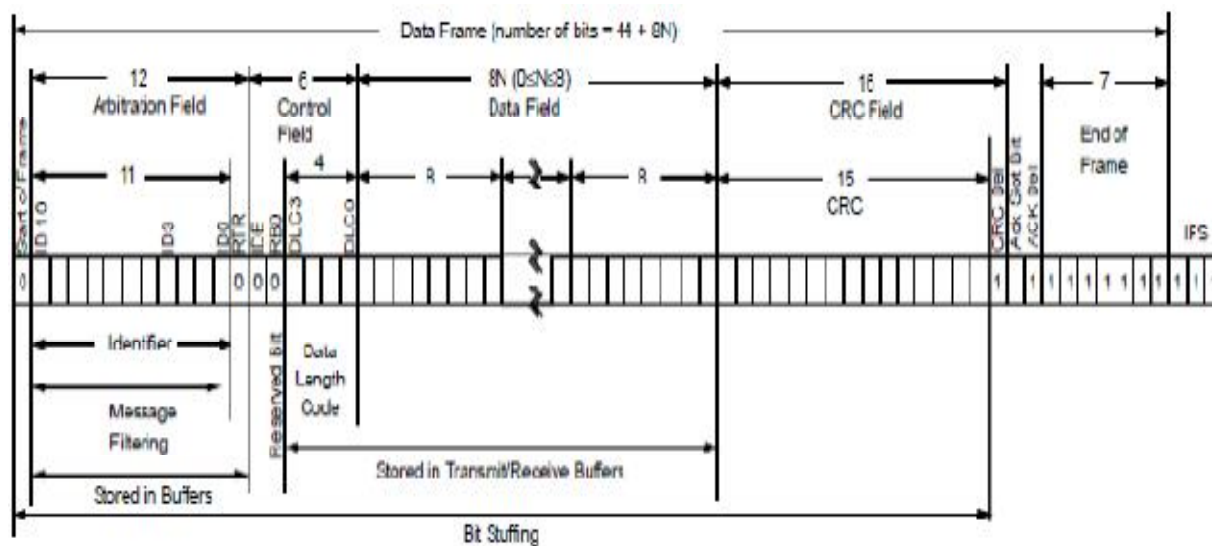
On the MCP1825 fixed output versions, the scaled-down output voltage is internally monitored and a power good (PWRGD) output is provided when the output is within 92% of regulation (typical). The PWRGD delay is internally fixed at 110 μ s (typical).

The overtemperature and short circuit current-limiting provide additional protection for the LDO during system fault conditions.



Pin	
1	V_{IN}
2	GND (IAB)
3	V_{OUT}
4	GND (TAB)

Anexo 9. Estructura del mensaje estandar



Anexo 10. Esquemáticos.

